



**Titre:** Problème de transport avec contraintes d'horaires  
Title:

**Auteur:** Nizar El Hachemi  
Author:

**Date:** 2009

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** El Hachemi, N. (2009). Problème de transport avec contraintes d'horaires [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie.  
Citation: <https://publications.polymtl.ca/198/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/198/>  
PolyPublie URL:

**Directeurs de recherche:** Louis-martin Rousseau, & Michel Gendreau  
Advisors:

**Programme:** Doctorat en mathématiques de l'ingénieur  
Program:

UNIVERSITÉ DE MONTRÉAL

PROBLÈME DE TRANSPORT AVEC CONTRAINTES D'HORAIRES

NIZAR EL HACHEMI

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION  
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR (Ph.D.)  
(MATHÉMATIQUES DE L'INGÉNIEUR)

DÉCEMBRE 2009

© Nizar El Hachemi, 2009.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

PROBLÈME DE TRANSPORT AVEC CONTRAINTES D'HORAIRES

présentée par : EL HACHEMI Nizar

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. SOUMIS, François, Ph.D., président

M. ROUSSEAU, Louis-Martin, Ph.D., membre et directeur de recherche

M. GENDREAU, Michel, Ph.D., membre et codirecteur de recherche

M. GENDRON, Bernard, Ph.D., membre

M. MICHEL, Laurent, Ph.D., membre externe

*À mes parents et à la mémoire de mes grands parents . . .*

## REMERCIEMENTS

Je tiens tout d’abord à exprimer vivement ma gratitude à l’égard de mon directeur de recherche Monsieur Louis-Martin Rousseau et à l’adresse de mon codirecteur Monsieur Michel Gendreau, pour m’avoir offert l’heureuse opportunité de réaliser ma thèse dans un domaine attrayant qui m’a autant donné de plaisir que stimulé et épanoui mon goût pour la recherche. Je suis particulièrement redevable à ces deux professeurs de qualité qui m’ont guidé avec une compétence qui n’a d’égal que leur dévouement à la science et à la noblesse de leur mission. Ils m’ont conseillé brillamment, soutenu financièrement et réconforté moralement, avec une rare générosité d’âme, toutes les fois que j’en avais besoin. Il faut bien dire que Louis-Martin Rousseau ne m’a guère laissé le temps de penser à l’inutile ni l’occasion de me fourvoyer, tant soit peu, dans l’incertain ou le flou.

Je remercie également Messieurs Jean Favreau et Joseph Nader de FPIInnovations qui m’ont fourni toutes les informations et toutes les données nécessaires à la définition du problème et qui étaient tout le temps disponibles et aimablement présents pour me clarifier les contraintes liées au contexte canadien.

Je me réserve le plaisir de louer l’aide incomparable dont j’ai bénéficié auprès des différentes composantes du personnel du CIRRELT : techniciens informatiques, secrétaires, étudiants et chercheurs.

Monsieur Laurent Michel, Monsieur François Soumis et Monsieur Bernard Gendron m'ont fait l'honneur de participer au jury de cette thèse. Je les en remercie avec une sincérité aussi profonde que pleine de respect. Cette thèse s'est fièrement parée de leurs noms pour signer sa mémoire et aborder, sans complexe, la durée.

Il m'est cher de remercier mes amis du CIRRELT et du GERAD et tout particulièrement Issmail El hallaoui dont la contribution à la réalisation de la troisième partie de ce travail a été hors de prix.

J'ai apprécié, par le cœur et l'esprit, le soutien sympathique de mon frère unique qui, malgré ses lourdes occupations professionnelles, a pu dégager une partie de son temps pour soulager mes efforts.

Enfin, aucun remerciement ne vaut ce que mes chers parents ont fait pour moi. Sans mon père et ma mère, ces deux habiles artisans de mes rêves, sans eux, sans leur accompagnement inconditionnel et assidu, rien de ce que j'ai réussi dans ma vie n'aurait été possible. C'est une chance providentielle que d'être leur enfant.

## RÉSUMÉ

L'industrie forestière est un secteur extrêmement important pour plusieurs pays dont le Canada. En 2007, ce secteur offrait de l'emploi à environ 1 million de personnes (directement et indirectement) à travers le pays et a contribué par \$23.4 milliards à la balance commerciale nationale. Plusieurs problèmes liés à cette industrie sont de nature d'aide à la décision. Ils se divisent généralement en trois catégories : stratégique, tactique et opérationnelle. Tout au long de cette thèse, nous nous sommes intéressé à cette dernière catégorie et plus précisément au problème du transport forestier avec horaire. Dans la littérature du domaine, cette question a fait l'objet de plusieurs travaux. De notre part, nous avons adapté le problème au contexte canadien en prenant en compte les contraintes de synchronisation entre les chargeuses et les camions. Ces contraintes de synchronisation traduisent le fait que les chargeuses en forêt ne peuvent pas supporter d'autres opérations en ces lieux, à part le chargement, vu la grande superficie des sites forestiers canadiens. Ainsi, il était primordial de minimiser les attentes des chargeuses et des camions, pour réduire les coûts de transport.

Dans le premier article de ce travail, nous avons traité le problème journalier où nous avons supposé que les requêtes de transport sont connues à l'avance. Une méthode hybride mettant en oeuvre la programmation par contraintes et la programmation linéaire en nombres entiers a été adoptée, de sorte que cette dernière modélise la

circulation des camions comme un problème de flot à coût minimum dans un réseau, alors que programmation par contraintes s'occupe de l'ordonnancement des tâches, une fois la circulation est établie. La communication entre les deux modèles se fait via des contraintes globales de cardinalité, permettant ainsi au modèle de la programmation par contraintes de profiter des algorithmes efficaces de filtrage de ces dernières.

Dans le deuxième article, nous avons abordé le problème hebdomadaire où nous avons pris en considération la politique de livraison en flux tirés juste à temps adoptée par les usines pour améliorer leurs gestions de stocks. Pour répondre à ce besoin, nous avons développé une méthode en deux phases. La première détermine les voyages pleins et les sites forestiers ouverts durant les jours de la semaine, en minimisant les coûts associés à ces deux activités. À la sortie de cette première phase, sept problèmes de transport journaliers sont générés, constituant la deuxième phase. Dans le but de résoudre les problèmes journaliers, un modèle de recherche locale basée sur les contraintes a été développé et résolu par l'intermédiaire d'un algorithme de recherche locale itérée. Ce dernier est basé d'une part sur cinq voisinages classiques en tournées de véhicules permettant l'optimisation de la partie routage, d'autre part sur un algorithme glouton développé spécialement pour effectuer l'ordonnancement des tâches, en tenant compte des contraintes de synchronisation. Dans certains cas, nous avons remarqué que l'algorithme glouton peut conduire à des coûts d'attente assez élevés. C'est pourquoi nous avons utilisé le modèle de programmation par contraintes de l'article précédent, donnant ainsi lieu à l'hybridation entre la recherche locale basée



sur les contraintes et la programmation par contraintes grâce à COMET1.1.

Finalement, le troisième article a traité, lui aussi, du problème hebdomadaire, mais en enrichissant le modèle journalier par d'autres contraintes pratiques. Nous avons donc considéré que les camions étaient affectés à des bases régionales à partir desquelles ils commençaient et finissaient leurs journées de travail et nous avons permis à chaque camionneur d'avoir une pause dîner entre  $11h$  et  $15h$ , chaque jour, dans n'importe quelle usine. Nous avons élaboré une méthode en deux phases dont la première est identique à celle de l'article précédent. Quant à la deuxième, un modèle de programmation linéaire en nombres entiers a été établi pour résoudre les problèmes journaliers. Il présente l'avantage d'avoir une structure très proche d'un problème de flot à coût minimum. L'autre avantage, c'est qu'il a permis d'obtenir des garanties sur la qualité de la meilleure solution entière obtenue.

## ABSTRACT

The forest industry is an important economic sector for several countries including Canada. In 2007, this industry employed about 1 million people (directly and indirectly), and contributed \$23.4 billion to Canada's trade balance. The operations research problems related to this sector are divided into three categories: strategic, tactical and operational. In this thesis, we are interested in the later category and more precisely in the log-truck scheduling problem. Many papers in the literature have addressed this issue, and our contribution has been to address the problem to the Canadian context, taking into account the synchronization constraints between loaders and trucks. These constraints reflect the fact that forest-loaders cannot support other operations in forests except loading, since in Canada, we have large areas.

In the first article of this thesis, we presented the daily problem where we have assumed that requests are known in advance. We proposed a hybrid approach involving a linear model to deal with the routing part of the problem and a constraint programming model to deal the scheduling part. Both of these models are combined through the exchange of global cardinality constraints.

In the second article, we discussed the weekly problem where inventories at wood mills are taken into consideration in order to allow wood mills to work in a just in

time mode. For this purpose, we have developed a two-phase method. The first one determines the loaded trips and opened forest areas during each day of the week. This phase yields seven daily problem, forming the second phase, where a constraint-based local search model have been developed. We solved these daily problems using an iterated local search algorithm based on five classical neighborhoods of the vehicle routing problem to deal with the routing part, whereas a greedy algorithm has been used to schedule activities (loading and unloading). In some cases, we notice that the greedy algorithm can lead to costly waiting time solutions. In this context, we adopted a hybrid method combining the constraint-based local search model and the previous constraint programming model of the first article.

Finally, the last paper addressed the weekly problem of the previous article with additional new practical constraints. We considered that trucks are affected to regional bases, so that each truck starts and ends its day at its related base, and we assured that each truck can have a break lunch between 11h and 15h each day at any wood mill. We developed a two-phase approach, the first one is identical to that of the preceding paper and the second phase is based on an integer linear model that is very close to a flow based model. The main advantage of this approach lies in the fact that it measures the quality of the best integer solution.

## TABLE DES MATIÈRES

|                                     |      |
|-------------------------------------|------|
| DÉDICACE . . . . .                  | iii  |
| REMERCIEMENTS . . . . .             | iv   |
| RÉSUMÉ . . . . .                    | vi   |
| ABSTRACT . . . . .                  | ix   |
| TABLE DES MATIÈRES . . . . .        | xi   |
| LISTE DES TABLEAUX . . . . .        | xvii |
| LISTE DES FIGURES . . . . .         | xix  |
| LISTE DES ANNEXES . . . . .         | xxi  |
| CHAPITRE 1 : INTRODUCTION . . . . . | 1    |

|  |           |
|--|-----------|
| 1.1 : Généralités . . . . .  | 1         |
| 1.2 : Outils d'aide à la décision et planification en foresterie . . . . .               | 3         |
| 1.3 : FPIinnovations . . . . .   | 5         |
| 1.4 : Présentation du problème . . . . .   | 6         |
| 1.5 : Approches de résolution . . . . .  | 11        |
| 1.5.1 : L'approche hybride programmation linéaire en nombres entiers<br>et PPC . . . . . | 13        |
| 1.5.2 : L'approche hybride recherche locale et PPC . . . . .                             | 14        |
| 1.5.3 : Troisième approche modèle PLNE . . . . .   | 16        |
| 1.6 : Organisation de la thèse . . . . .   | 18        |
| <b>CHAPITRE 2 : REVUE DE LITTÉRATURE . . . . .</b>                                       | <b>20</b> |
| 2.1 : Problème de tournées de véhicules . . . . .  | 21        |
| 2.2 : Transport en milieu forestier . . . . .  | 25        |
| 2.2.1 : Planification tactique du transport de bois . . . . .                            | 27        |

|   |    |
|---|----|
| 2.2.2 : Problème d'horaire du transport des billes de bois (LTSP) . | 28 |
| 2.3 : Synchronisation . . . . .                                     | 39 |

### **CHAPITRE 3 : A HYBRID CONSTRAINT PROGRAMMING APPROACH TO THE LOG-TRUCK SCHEDULING PROBLEM . . . . .**

|  |    |
|--|----|
| 3.1 : Introduction . . . . .                         | 45 |
| 3.1.1 : The Log-Truck Scheduling Problem . . . . .   | 45 |
| 3.1.2 : Literature Review . . . . .                  | 47 |
| 3.1.3 : A Constraint Programming Approach . . . . .  | 49 |
| 3.2 : Problem Description and Model . . . . .        | 50 |
| 3.2.1 : Parameters . . . . .                         | 51 |
| 3.2.2 : Variables and domains . . . . .              | 51 |
| 3.2.3 : Constraints and Objective Function . . . . . | 52 |
| 3.3 : Solving the LTSP . . . . .                     | 54 |

|  |               |
|--|---------------|
| 3.3.1 : Decomposing the LTSP . . . . .   | 55            |
| 3.3.2 : IP Model, variables, parameters and constraints . . . . .                                      | 57            |
| 3.3.3 : Exploring other IP solutions . . . . .   | 58            |
| 3.4 : Experimental Results . . . . .   | 61            |
| 3.5 : Conclusion . . . . .   | 69            |
| <br><b>CHAPITRE 4 : A HEURISTIC TO SOLVE THE SYNCHRONIZED<br/>LOG-TRUCK SCHEDULING PROBLEM . . . .</b> | <br><b>70</b> |
| 4.1 : Introduction . . . . .   | 74            |
| 4.2 : Literature review . . . . .  | 75            |
| 4.3 : Problem description . . . . .  | 78            |
| 4.4 : Solution Approach . . . . .  | 79            |
| 4.5 : The Tactical Model . . . . .   | 80            |
| 4.5.1 : Parameters . . . . .   | 81            |
| 4.5.2 : Variables of the Model . . . . .   | 81            |

|   |                |
|---|----------------|
| 4.5.3 : Constraints and Objective . . . . .   | 81             |
| 4.5.4 : Solution Method . . . . .   | 82             |
| 4.6 : The Daily Synchronized Log-Truck Scheduling Problem . . . . .   | 84             |
| 4.6.1 : A Constraint-Based Local Search Model . . . . .   | 86             |
| 4.6.2 : A Hybrid Approach . . . . .   | 92             |
| 4.7 : Experimental Results . . . . .  | 97             |
| 4.8 : Conclusion . . . . .  | 102            |
| <br><b>CHAPITRE 5 : AN IMPLICIT INTEGER LINEAR PROGRAM<br/>TO SOLVE THE WEEKLY LOG-TRUCK SCHE-<br/>DULING PROBLEM . . . . .</b> | <br><b>103</b> |
| 5.1 : Introduction . . . . .  | 107            |
| 5.2 : Log-Truck Scheduling Problem . . . . .  | 108            |
| 5.3 : Literature Review . . . . .   | 109            |
| 5.4 : Solution Approach . . . . .   | 112            |



|   |            |
|---|------------|
| 5.4.1 : The Tactical Problem . . . . .  | 113        |
| 5.4.2 : Phase 2 : The Daily Synchronized Log-Truck Scheduling Problem . . . . . | 115        |
| 5.5 : Experimental Results . . . . .  | 122        |
| 5.6 : Conclusion . . . . .  | 125        |
| <b>CHAPITRE 6 : CONCLUSION . . . . .</b>  | <b>132</b> |
| 6.1 : Limites et difficultés de l'étude . . . . .                               | 134        |
| 6.2 : Directions de recherches . . . . .  | 136        |
| <b>BIBLIOGRAPHIE . . . . .</b>  | <b>136</b> |
| <b>ANNEXES . . . . .</b>  | <b>148</b> |

# LISTE DES TABLEAUX

|   |     |
|---|-----|
| Table 3.1 : <b>Results of the first case study</b> . . . . .                                      | 62  |
| Table 3.2 : <b>Results of the second case study</b> . . . . .                                     | 63  |
| Table 3.3 : <b>Results of the first case study for six different computation times</b> . . . . .  | 65  |
| Table 3.4 : <b>Results of the second case study for six different computation times</b> . . . . . | 66  |
| Table 3.5 : <b>Decomposition Approach with perturbation <math>c \leq 30</math></b> . .            | 67  |
| Table 4.1 : <b>Results of case 1</b> . . . . .  | 99  |
| Table 4.2 : <b>Results of case 2</b> . . . . .  | 100 |
| Table 4.3 : <b>The best total cost results</b> . . . . .  | 101 |
| Table 4.4 : <b>The variance results</b> . . . . .   | 101 |
| Table 5.1 : <b>Results with one base and 20 minutes step discretization</b>                       | 126 |

Table 5.2 : **Results with one base and 10 minutes step discretization** 127

Table 5.3 : **Results with three bases and 20 minutes step discreti-**  
**zation** . . . . . 128

Table 5.4 : **Results with three bases and 10 minutes step discreti-**  
**zation** . . . . . 129

Table 5.5 : **Results with a base for each truck and 20 minutes step**  
**discretization** . . . . . 130

## LISTE DES FIGURES

|  |    |
|--|----|
| Figure 1.1 : Exemple d'un voyage de camion . . . . .                           | 11 |
| Figure 1.2 : Première approche . . . . .                                       | 14 |
| Figure 1.3 : Deuxième approche . . . . .                                       | 17 |
| Figure 2.1 : Exemple de tournées de deux véhicules . . . . .                   | 22 |
| Figure 2.2 : Tours simples et complexes . . . . .                              | 28 |
| Figure 3.1 : Graph associated with the empty driving . . . . .                 | 57 |
| Figure 3.2 : Results of the first scenario of the first case study . . . . .   | 68 |
| Figure 3.3 : Results of the second scenario of the second case study . . . . . | 68 |
| Figure 4.1 : Network example of tactical model . . . . .                       | 84 |
| Figure 4.2 : First part of the tactical problem algorithm . . . . .            | 85 |

|  |     |
|--|-----|
| Figure 4.3 : Second part of the tactical problem algorithm . . . . . | 85  |
| Figure 4.4 : The second phase hybrid algorithm . . . . .             | 96  |
| Figure 5.1 : Example of a route for a logging truck . . . . .        | 110 |
| Figure 5.2 : Network Arcs Formulation Model . . . . .                | 119 |
| Figure 5.3 : Enriched Network Model . . . . .                        | 120 |
| Figure 5.4 : Branching Strategy . . . . .                            | 122 |

## **LISTE DES ANNEXES**

|  |            |
|--|------------|
| <b>ANNEXE A : MODÈLES EN COMET ET IMPLEMENTATION .</b> | <b>148</b> |
|--|------------|

# CHAPITRE 1 : INTRODUCTION

## 1.1 Généralités

Tout au long de l'histoire humaine, les forêts ont représenté une source de vie, d'alimentation et d'énergie. De nos jours, l'industrie forestière, occupe une place importante dans plusieurs pays du monde tels le Chili, le Canada, la Suède, la Finlande, la Nouvelle-Zélande et l'Autriche, entre autres. Le Canada possède 417 millions d'hectares de forêts (voir Ressources Naturelles Canada [2005]) et dispose d'une industrie forestière très importante offrant à peu près 1 million d'emplois dont 255000 directs et 763000 indirects. En 2007, la contribution de cette activité économique à la balance commerciale du Canada est de 23,4 milliards de dollars. En Suède, par exemple, 60% du territoire est couvert par des forêts et, selon le National Board of Forestry (Skogsstyrelsen), l'industrie forestière représente 20% du total des recettes d'exportation en 2003.

Actuellement, les demandes des clients sont devenues de plus en plus variées en terme de la qualité, de la taille et des types de bois sollicités. Le temps est devenu un facteur très important dans les livraisons, surtout quand il s'agit de bois de haute qualité, dont il faut garder la fraîcheur et éviter les ruptures de stocks ou de bois à pâte qui perd rapidement sa fraîcheur, nécessitant par là même un stockage minimal. Outre les

contraintes liées aux exigences dictées par les clients, la préservation de ces espaces boisés, de la faune et des espèces animales et végétales s'impose en tant que contrainte écologique visant la protection de la forêt considérée comme poumons de la terre et filtre absorbant le dioxyde de carbone à l'échelle de la planète.

L'industrie forestière dépend dans plusieurs pays, et en particulier au Canada, des marchés d'exportation. Ainsi, la plus grande partie de la production canadienne est vendue à l'extérieur. Ces exportations sont principalement acheminées vers les États-Unis, l'Union européenne et le Japon. Dans un contexte commercial international où la concurrence bat son plein, la position de l'industrie forestière Canadienne n'est pas à l'abri d'un péril qu'elle doit éviter. Pour toutes ces raisons (place de l'industrie forestière dans l'économie, rôle écologique des forêts, contraintes et restrictions relatives aux clients), les compagnies forestières qui aspirent à la réussite n'auront d'autre choix que de se mettre au diapason des exigences du marché et des impératifs écologiques, en se dotant d'une gestion plus efficiente. Aussi l'étroite collaboration avec les chercheurs en recherche opérationnelle (RO) s'avère-t-elle cruciale eu égard à cette industrie, dans la perspective de lui fournir des outils d'aide à la décision qui lui permettront d'augmenter la valeur de son offre et de diminuer ses coûts, tout en ayant une vision stratégique à long terme respectant les restrictions écologiques et offrant une exploitation durable dans le temps. Nous décrirons par la suite certains problèmes classiques inhérents à cette industrie et, en particulier, la démarche de planification des activités d'une compagnie forestière.



## 1.2 Outils d'aide à la décision et planification en foresterie

Les outils de recherche opérationnelle sont largement utilisés dans plusieurs secteurs et applications forestières depuis une trentaine d'années environ. Or, plusieurs problèmes liés à l'industrie forestière ont été traités grâce à la RO. De ce fait, des prix tels celui d'Edelmann (voir Epstein et al. [1999]), reconnaissent le rôle et l'impact de la RO en foresterie. On présentera par la suite une liste non exhaustive de ces problèmes que nous classons ci-après en trois catégories :

- Problèmes liés aux forêts uniquement
- Problèmes associés aux usines uniquement
- Problèmes associés à la fois aux forêts et aux usines

La première classification inclut une liste très large de problèmes (construction et conception de routes, gestion des incendies (voir Martell [1982] et Martell et al. [1998]), gestion des surfaces boisées à exploiter (voir Epstein et al. [2007])). La deuxième classification comporte des problèmes classiques, comme celui de la découpe de rouleaux en usines à pâtes de papiers dans le but de minimiser les pertes (voir Chauhan et al. [2008]), celui de l'optimisation du processus de contrôle de la production, d'ouverture ou de fermeture d'usines, .... La dernière classification représente tous les problèmes liant les forêts et les usines, par exemple le choix des moyens et technologies de transport (camions, remorques, trains, etc), la planification des

routes et de l'horaire de camions, .... Dans cette thèse, nous nous intéressons à cette dernière catégorie de problèmes, spécialement le transport entre forêts et usines. Les paragraphes suivants décrivent les trois parties de la planification en foresterie ; stratégique, tactique et opérationnelle.

La planification stratégique en foresterie consiste à prendre des décisions aux niveaux de la gestion des surfaces boisées à exploiter, de la construction des routes, de l'ouverture ou de la fermeture d'usines, des processus d'investissements (choix de machines, moyen de transport, etc...) et finalement au niveau du choix de la politique de gestion de production ("make-to-stock", "make-to-order", "cut-to-order"). Ces politiques peuvent affecter d'autres décisions d'investissements, comme par exemple les capacités de stockage et les types d'équipements aux usines qui, pour supporter une politique make-to-stock, diffèrent de ceux de la politique make-to-order. En général, la planification stratégique s'étend sur de longues périodes étalées sur plusieurs années.

La planification tactique s'étend sur des horizons moyennement longs allant selon le type de problème traité, de 1 à 5 ans. Parmi ces problèmes, on trouve essentiellement l'allocation de ressources dans les différentes parties de la chaîne logistique en forêt, l'évaluation et l'estimation des budgets annuels. Les décisions tactiques représentent un pont de liaison reliant les décisions stratégiques et opérationnelles de manière à ce que ces dernières contribuent efficacement à la réalisation des directives stratégiques.

La planification opérationnelle est la dernière étape de contrôle dans la planification de la chaîne logistique en foresterie. Elle couvre de très courtes durées variant de quelques heures à quelques jours. Dans cette catégorie, on rencontre des problèmes très variés tels le découpage de bois, l'ordonnancement de la chaîne de production en temps réel, le transport et la répartition du bois entre les forêts et les usines, entre autres. À ce niveau de la planification, l'application pratique doit comporter le plus possible de détails pour que les décisions soient utilisables sans modification.

### **1.3 FPInnovations**

FPInnovations a été créé en avril 2007, à partir de la fusion de FERIC, Forintek, Paprican et du Centre canadien sur la fibre de bois. Il constitue le plus important institut au monde à but non lucratif oeuvrant en foresterie avec plus de 600 employés répartis à travers le Canada.

FERIC (Institut canadien de recherches en génie forestier) a été créé en 1975 dans le but d'améliorer et d'optimiser les opérations forestières canadiennes au profit de ses différents partenaires qui, eux, financent ses recherches en partenariat à la fois avec le gouvernement fédéral et avec les gouvernements provinciaux. Aujourd'hui, FERIC réunit un effectif de plus de 100 employés en service essentiellement dans les bureaux de Montréal, Québec et Vancouver. En conformité avec la vocation de ces centres, la définition des problèmes et les données associées à cette thèse ont été fournies par le bureau du FERIC à Montréal.

Forintek (Institut national de recherche sur les produits du bois au Canada) a été créé en 1915 dans le but de soutenir l'industrie forestière en matière d'optimisation des procédés de fabrication et d'extraction de valeur de la matière première, grâce au financement de ses membres et des gouvernements provinciaux. Actuellement, l'institut compte environ 170 professionnels reconnus à l'échelle internationale pour leurs réalisations.

Paprican (Institut canadien de recherches sur les pâtes et papiers) a été créé il y a plus de 80 ans. Ses programmes de recherche sont axés sur l'industrie des pâtes et papiers, lui offrant des technologies de grande valeur qui répondent aux besoins de ses membres et de ses partenaires.

Enfin, pour terminer, vient le centre canadien sur la fibre de bois (CCFB) regroupe des chercheurs oeuvrant dans le secteur forestier, s'intéressant précisément à la fibre ligneuse. L'objectif de ces chercheurs consiste en la production de connaissances innovatrices aidant l'industrie du bois à tirer profit des fibres ligneuses.

## 1.4 Présentation du problème

Plusieurs motivations poussent l'industrie forestière à améliorer sa productivité et à minimiser ses coûts. À ce sujet, la pression exercée par la concurrence mondiale, les réglementations environnementales concernant l'émission de gaz à effet de serre et les exigences de plus en plus prononcées des clients en termes de services sont les principaux déclencheurs d'une politique efficace de gestion et de planification.

Au Québec, au sein du processus intégral de fabrication d'un produit en bois, le transport représente un peu plus de 30% des coûts d'approvisionnement en bois. Ces coûts semblent être compressibles, surtout lorsqu'on constate que 50% des coûts de carburants sont liés au transport sans charge utile. Le transport du bois au Canada se déroule en deux étapes séparées. La première est dite transport primaire. Elle consiste à acheminer le bois de la forêt vers le bord du chemin à l'aide de débardeurs, porteurs ou autres machines forestières, tandis que la deuxième phase effectue le transport de bois des bords des chemins vers les usines à l'aide de camions remorques, semi-remorques, trains et autres. Actuellement, la planification du transport forestier au Canada est assurée manuellement par des experts dans la matière (contre-maîtres). Cependant, l'augmentation des volumes de bois à transporter et la complexité des contraintes propres au problème rendent la résolution manuelle difficile et longue.

Certains pays ont été leaders en matière de développement d'outils ou de logiciels d'aide à la décision destinés à résoudre la problématique du transport (journalier ou hebdomadaire) entre les forêts et les usines. Le Chili et la Finlande ont mis en oeuvre des outils appelés respectivement ASICAM et EPO dès le début des années 90 (voir Weintraub et al. [1996] et Linnainmaa et al. [1995]). Dans la même période, Rönnqvist et Ryan [1995] ont élaboré un outil de répartitionnement en temps réel adapté à l'industrie forestière en Nouvelle-Zélande. Par la suite, la Suède a développé des programmes de gestion du transport journalier en forêt par le biais des travaux de Palmgren et al. [2003], Palmgren et al. [2004] et Palmgren [2005] et un peu plus tard grâce aux travaux de Flisberg et al. [2009] et Andersson et al. [2008]. Au Canada,

l'utilisation des outils d'aide à la décision en transport forestier est très limitée pour ne pas dire inexistante, mis à part le projet VTM réalisé grâce à la collaboration de FORAC et FERIC. Ceci nous a motivé à travailler sur la deuxième phase du transport entre les bords des chemins forestiers et les usines. Par conséquent, dans cette thèse, nous nous sommes intéressés en premier lieu au problème journalier. Ensuite, nous avons étendu l'horizon de planification à une semaine, en prenant en compte la politique de livraison juste à temps souhaitée par les usines, pour répondre à leurs besoins journaliers, sans trop les dépasser, pour une meilleure gestion des stocks.

Pour mieux saisir l'importance du problème type canadien défini par FPInnovations (FERIC) sur lequel nous nous sommes appuyés dans notre thèse, il s'avère utile d'évoquer tout d'abord le problème connu dans la littérature sous le terme "Log-Truck Scheduling Problem" (LTSP). Celui-ci consiste à planifier l'horaire des camions en avance, pour satisfaire la demande, tout en minimisant les coûts associés. Une deuxième alternative s'offre pour résoudre le problème journalier. Elle se propose de planifier l'horaire des camions de façon continue en temps réel, tout au long de la journée. L'avantage de cette approche se traduit par le fait de pouvoir gérer l'impact des variations brusques qui surviennent au cours du plan à cause des pannes, du climat, des coupures de routes et d'autres facteurs du même ordre. Le LTSP présente beaucoup de similitude avec certains problèmes étudiés dans la littérature, en l'occurrence, le "Vehicle Routing Problem" (VRP) et le "Pick-up and Delivery Problem with Time Windows" (PDPTW). Cependant, un LTSP classique est caractérisé par

plusieurs contraintes et facteurs, tel le fait que les camions ont leurs propres bases à partir desquelles ils commencent et finissent leurs journées. Chaque camion a sa propre capacité de chargement et son horaire de travail spécifique. En général, deux types de camions sont considérés, ceux avec et ceux sans chargeuse. Les repos ou changements de chauffeurs sont pris en compte lorsqu'une route excède la limite de temps de conduite pour un chauffeur. La quantité de bois disponible aux bords des chemins des forêts ou celle exprimée comme demandes aux usines n'est pas forcément un multiple de chargement total d'un camion. Les coûts du transport à vide et plein sont pris en compte, alors que les chargeuses en forêt ont un quart de travail fixe pendant la journée.

Dans notre étude, plusieurs hypothèses ont été considérées. Certaines hypothèses ont facilité le problème par rapport à un LTSP standard, d'autres l'ont compliqué un peu. Mise à part la contrainte de repos ou de changement de chauffeurs qui n'a pas été traitée au cours de ce projet de recherche, toutes les autres contraintes reposent sur la réalité de terrain du contexte forestier canadien. Ainsi, les descriptions qu'on a reçues de FPInnovations (FERIC) indiquent que les camions sont homogènes, d'une capacité de 32 tonnes chacun et sans auto-chargeuse. Les camions n'ont pas forcément de bases dans la majorité des cas étudiés, puisqu'ils peuvent effectuer les changements de chauffeurs, commencer et finir leurs journées dans n'importe quelle usine. La structure des coûts associée aux camions est constituée de trois composantes : les coûts des voyages à vide, les coûts des voyages pleins et finalement ceux des attentes. Dans la pratique, les demandes aux usines et la quantité de bois en

forêts sont généralement exprimées en unité de chargement plein (32 tonnes dans notre cas). Ceci est dû au fait que les surfaces boisées exploitées au Canada sont très larges et engendrent de grands volumes de bois facilement convertissables en unité de chargement plein. Les erreurs dans l'estimation de la demande sont amorties par la capacité de stockage journalier dans les usines. Par rapport à l'utilisation des véhicules, un voyage de camion est composé d'un aller-retour (voir figure 1.1) usine de départ- forêt de chargement- usine de destination. Par conséquent, le chargement total du camion (32 tonnes) s'effectue à la forêt associée au voyage. Les chargeuses en forêts n'ont pas de quarts de travail fixes, leurs horaires sont intégrés à la problématique de manière à minimiser leurs coûts d'attentes. Ceci est différent d'un LTSP classique dans lequel les chargeuses travaillent des quarts de travail fixes, car elles sont supposées supporter d'autres opérations dans les forêts pendant leurs temps d'attentes. Dans le contexte canadien, les chargeuses ne peuvent pas contribuer à d'autres opérations (rangement de piles de bois, etc), du fait que, comme expliqué auparavant, les surfaces boisées sont très étendues et que chaque déplacement de la chargeuse entre les piles de bois peut dépasser les temps séparant l'arrivée de deux camions qui se succèdent. En effet, minimiser les coûts d'attentes des camions et des chargeuses permet d'augmenter leurs taux d'utilisation et, en conséquence, la rentabilité de la compagnie. À notre connaissance, à part le travail de Bredström and Rönnqvist [2008] où la synchronisation entre les camions et les chargeuses a été prise en compte, cette thèse contribue à ce niveau, en intégrant systématiquement l'horaire des chargeuses à celui des camions pour un horizon journalier en premier lieu, hebdomadaire en second lieu. Tout cela en tenant compte des coûts d'attentes



des deux ressources (camions et chargeuses). Nous verrons par la suite les différentes approches développées pour résoudre le problème.

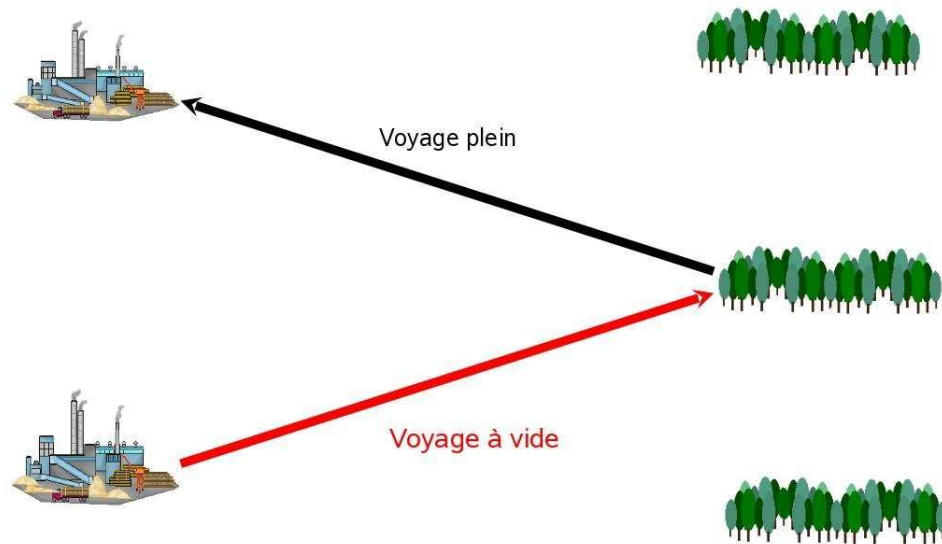


Figure 1.1 – Exemple d’un voyage de camion

## 1.5 Approches de résolution

Trois modèles différents sont proposés dans cette thèse pour résoudre ce qu’on a appelé le “Synchronized Log-Truck Scheduling Problem” (SLTSP), où on synchronise

les camions avec les chargeuses, pour minimiser la somme de leurs coûts d'attentes.

Un premier modèle de type de programmation par contraintes (PPC) a été élaboré pour traiter le problème journalier. La PPC se divise en deux parties : premièrement, la modélisation qui est dite déclarative et qui se compose essentiellement de variables de décision appartenant à des domaines finis liés par des contraintes mathématiques ou symboliques ; deuxièmement, la résolution qui se déroule en deux phases : propagation de contraintes et stratégies de recherche. La propagation de contraintes est un mécanisme d'inférence visant à réduire les configurations incompatibles. En général, elle est incomplète, puisque les valeurs qui restent dans les domaines des variables peuvent être incompatibles d'un point de vue global. Ainsi, la recherche est nécessaire pour explorer le reste de l'arbre. Ces deux phases de la résolution (propagation de contraintes et recherche) sont combinées, puisqu'après chaque décision de branchement, la propagation est à nouveau déclenchée. L'avantage crucial que présente la PPC réside dans la facilité de modélisation qu'elle offre aux chercheurs et dans la richesse des bibliothèques de contraintes des différents solveurs (voir Baptiste et al. [2001] et Milano [2004]).

Le deuxième modèle est de type recherche locale basée sur les contraintes (RLBC). Cette nouvelle programmation est supportée par le solveur COMET et combine une modélisation héritée de la PPC avec des heuristiques de recherche locale. Le rôle de la RLBC consiste à la fois à modéliser les problèmes de façon déclarative et à guider la recherche locale en se basant sur les contraintes et l'objectif. Plusieurs aspects du

SLTSP ont été modélisés par ce qu'on appelle des invariants (variables incrémentales). Ces invariants sont maintenus automatiquement par COMET.

Le troisième et dernier modèle est un programme linéaire en nombres entiers (PLNE) qui possède une structure très proche d'un problème de flot à coût minimum dans un réseau. L'avantage crucial de ce modèle réside dans le fait que le nombre de contraintes est indépendant du nombre des lots à transporter, puisque la gestion des chargeuses est assurée par la structure de flot et les capacités associées aux arcs de chargements et déchargements du modèle.

### **1.5.1 L'approche hybride programmation linéaire en nombres entiers et PPC**

La première méthode développée dans cette thèse traite uniquement du problème journalier. Cette approche propose une hybridation naturelle entre la PLNE et la PPC, sachant que la première s'occupe de la partie tournée des camions, alors que la PPC se charge de l'ordonnancement des tâches. Trois avantages essentiels découlent de cette méthode : premièrement, la synchronisation des camions avec les chargeuses se fait systématiquement par le biais du modèle déclaratif PPC proposé ; deuxièmement, le modèle PLNE considéré est un simple problème de flot à coût minimum dans un réseau qu'on résout avec Cplex. Finalement, la communication entre les deux modèles se fait via l'introduction des contraintes globales de cardinalités (GCC). Ces dernières transmettent au modèle PPC la tournée optimale des camions tout en ayant

des algorithmes efficaces de propagation (voir figure 1.2)

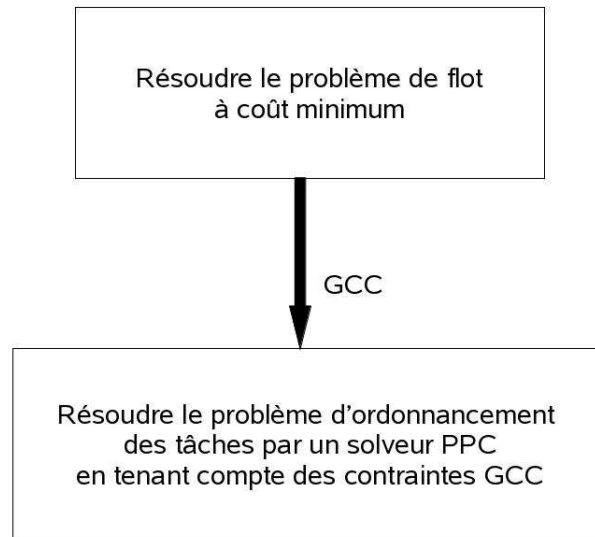


Figure 1.2 – Première approche

### 1.5.2 L'approche hybride recherche locale et PPC

Dans cette deuxième méthode, on s'intéresse au problème hebdomadaire dans lequel les usines adoptent une politique de livraison en flux tirés juste à temps. En effet, les usines essaient de ne pas trop dépasser leurs demandes journalières de manière à avoir

une bonne gestion des stocks. Cette approche se décompose en deux phases séquentielles. La première, qu'on a appelée phase tactique, permet de satisfaire la demande des usines, en respectant les contraintes de capacité des chargeuses, les contraintes de stocks intermédiaires et finalement les contraintes qui assurent un fonctionnement minimal de chaque chargeuse une fois celle-ci en marche. Cette phase a été résolue approximativement par un algorithme tabou codé en COMET. Entre décembre et février, la production s'intensifie en forêt en prévision de l'arrêt de printemps dû au dégel. Entre mars et mai, il y a une période d'environ quatre à six semaines durant laquelle aucun travail n'est possible en forêt puisque le sol est trop mou. Les usines doivent avoir entreposé assez de bois pour travailler pendant ces semaines. Ce qui arrive en décembre, c'est que les usines gardent la même politique de livraisons ("just-in-time mode"), alors que les forêts ont déjà accentué leurs productions. Pour cette raison, nous ne tenons pas compte de la contrainte de production en forêt dans notre modèle de la phase tactique durant la période de décembre. Une autre façon d'appliquer notre modèle tactique (sans contraintes de production) consiste à planifier l'entreposage de bois dans les zones de stockage (entre décembre et février) pour subvenir aux besoins d'usines pendant la période de dégel ; mais à ce moment, il faudrait ignorer la politique du "just-in-time".

La deuxième phase représente sept SLSPS journaliers couvrant toute la semaine : du lundi au dimanche. Nous les résolvons séquentiellement dans l'ordre de la semaine. Deux modèles ont été développés pour traiter le SLTSP journalier. Le premier est un modèle PPC identique à celui présenté dans le chapitre 2, à quelques détails près.

Le second est un modèle RLBC basé sur une majorité de variables incrémentales, à part le temps du début de chargement, le temps du début de déchargement et les variables ordonnant les lots par camion. La partie tournée de camions est résolue par une recherche locale basée sur cinq structures de voisinage différentes, alors que la partie ordonnancement a été résolue par deux modules dépendant du type de modèle. Ainsi, un algorithme glouton ordonnant les tâches au plus tôt possible est utilisé pour la partie RLBC, alors que dans l'autre partie, le solveur PPC de COMET s'occupe de l'ordonnancement. La PPC s'avère parfois être une très longue méthode, surtout si on la compare à l'algorithme glouton dans notre cas. Certes, la qualité des solutions trouvées par la PPC est relativement meilleure, comparée à celle de l'algorithme glouton. Cependant, le critère du temps global alloué à chaque problème journalier est très contraignant. C'est pour cette raison que, le modèle PPC a été hybridé avec celui de la RLBC dans des situations différentes bien précises qui nous semblent potentiellement prometteuses et qui ont été testées indépendamment (voir figure 1.3).

### **1.5.3 Troisième approche modèle PLNE**

Dans cette partie, nous avons affaire à la même stratégie tactique adoptée dans la deuxième contribution, avec une petite modification : nous considérons les contraintes de stocks de bois en forêts. Cependant, comme auparavant, la deuxième phase est constituée de sept SLTSPS journaliers. Nous proposons un modèle en nombres entiers adapté au problème journalier et enrichi par rapport à ceux des deux premières contributions, en ajoutant une contrainte de pause repas par camion aux usines entre

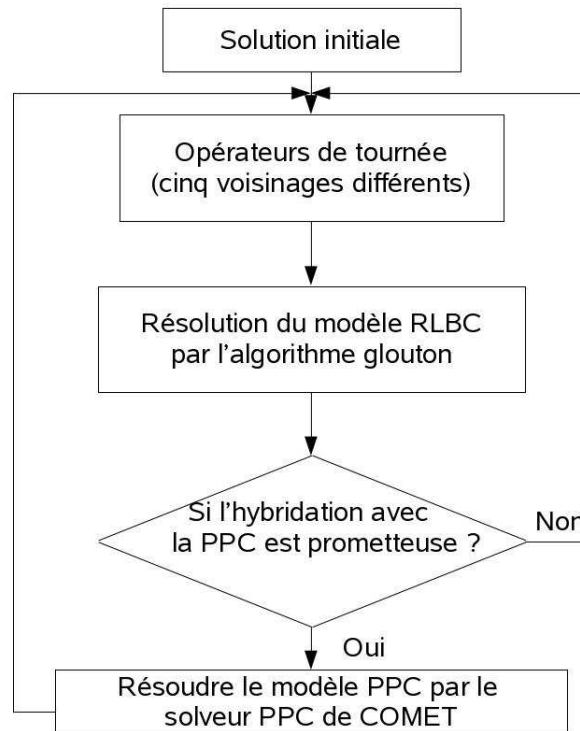


Figure 1.3 – Deuxième approche

11h et 15h. Le modèle PLNE proposé a une structure très proche d'un problème de flot dans un réseau. Pour cette raison, nous obtenons assez rapidement des résultats de bonne qualité comparativement à ceux des autres modèles, tout en ayant des garanties d'optimalité très satisfaisantes. En outre, l'avantage crucial que présente ce modèle par rapport aux autres consiste dans le fait que le nombre de contraintes est indépendant du nombre des lots à transporter.

## 1.6 Organisation de la thèse

Cette thèse est organisée en six chapitres. Le premier présente le contexte forestier canadien, l'utilité de l'emploi des méthodes de la RO en foresterie, une présentation générale du centre de recherche partenaire de notre équipe, la problématique étudiée ainsi que les différentes approches développées dans le cadre de ce travail. Le deuxième porte sur une revue de la littérature traitant des différentes méthodes élaborées dans le cadre du LTSP et ses variantes. Le troisième se consacre à la présentation de notre première contribution qui s'applique à résoudre le problème journalier en hybridant la PLNE avec la PPC. Le quatrième traite du problème hebdomadaire en prenant en considération la livraison en flux tirés juste à temps adoptée par les usines. Nous y développons une méthode heuristique pour résoudre ce problème. Cette dernière combine la RLBC et la PPC de manière à tirer profit de la rapidité des heuristiques et de l'efficacité de la PPC dans les problèmes d'ordonnancement. Le cinquième chapitre se concentre sur la description d'une nouvelle approche destinée à résoudre une version enrichie du problème hebdomadaire permettant de considérer la contrainte de la production en forêt et les pauses attribuées aux camions en usines entre 11h et 15h, chaque jour. En effet, une nouvelle modélisation en nombres entiers du SLTSP très proche de la structure d'un problème de flot dans un réseau permet d'avoir des solutions de meilleure qualité comparativement à ceux du chapitre 4. En outre, cette nouvelle modélisation assure l'indépendance entre le nombre de contraintes et le nombre de lots à transporter. Le sixième et dernier chapitre donne lieu à un bilan



qui reprend les phases importantes du projet de recherche et souligne les différentes contributions et les limites de celles-ci, en concluant sur les pistes et avenues de recherches soulevées par ce travail.

## CHAPITRE 2 : REVUE DE LITTÉRATURE

On présentera essentiellement dans ce chapitre, une revue de littérature des différentes approches élaborées en gestion du transport forestier. On abordera, en premier lieu, des problèmes classiques de transport, comme celui des tournées de véhicules avec ou sans fenêtre de temps (PTVFT, PTV) ou celui de chargement/déchargement avec fenêtre de temps (PCDFT). Le but de cette introduction générale aux problèmes standards de transport est d'établir les liens et points de ressemblance et de différence entre ces problèmes et le problème de transport en contexte forestier (PTCF). En deuxième lieu, on s'intéressera à plusieurs approches adaptées au PTCF depuis la fin des années 80. On se focalisera spécialement sur les problèmes intégrant la composante du temps (LTSP) et, partant, on décrira plusieurs méthodes de planification des horaires de camions développées dans différents pays. Par la suite et dans la même direction, on abordera les principales approches de planification d'horaire de camions en temps réel.

La contribution principale de cette thèse consiste à résoudre le SLTSP en construisant à la fois l'horaire des camions et celui des chargeuses en forêts et usines, en synchronisant ces deux ressources (camions et chargeuses) pour réduire au maximum les coûts d'attentes sans pour autant oublier les coûts de transport. Par conséquent, une partie de cette recension des écrits portera sur les problèmes et applications où la synchronisation joue un rôle fondamental comme la planification d'horaire d'autobus

et de chauffeurs, la planification des visites des équipes de soins de santé à domicile, entre autres.

## 2.1 Problème de tournées de véhicules

Le PTV est l'un des problèmes classiques du transport qui dérive du problème du voyageur de commerce, avec en plus la possibilité de visiter l'ensemble des villes par plusieurs véhicules, dont on doit tenir compte de la capacité de chacun (voir figure 2.1). Ce problème a été introduit pour la première fois par Dantzig and Ramser [1959] dans le cadre d'une étude de distribution d'essence. Plusieurs modèles et formulations de la littérature ont traité le PTV. On décrira dans la suite deux formulations, la première (P) est dite basée sur les contraintes de partitionnement ou modèle génération de colonnes, la deuxième (C) est compacte de type PLNE proposée par Golden et al. [1997].

$$P = \text{Min} \sum_{j \in \Omega} c_j x_j$$

sujet à :

$$\sum_{j \in \Omega} a_{ij} x_j = 1, \quad \forall i \in V. \quad (2.1)$$

$$\sum_{j \in \Omega} x_j \leq n, \quad . \quad (2.2)$$

$$x_j \in \{0, 1\}, \quad \forall j \in \Omega. \quad (2.3)$$

$x_j$  est une variable binaire représentant une colonne qui correspond à une route possible d'un véhicule, de sorte que si la colonne est sélectionnée  $x_j$  vaut 1, sinon 0.

$a_{ij}$  est une constante de valeur 1 si la colonne (route)  $j$  visite la ville  $i$  appartenant à l'ensemble des villes  $V$ .  $\Omega$  est l'ensemble des routes (colonnes) réalisables (vérifiant que  $\sum_{i \in V} a_{ij} d_i \leq q$ ),  $d_i$  étant la demande associée à la ville  $i$  et  $q$  la capacité d'un véhicule quelconque des ( $n$  nombre de camions) disponibles.  $c_j$  est le coût associé à la route  $j$ . Au cours du processus de résolution, l'ensemble  $\Omega$  est généré au fur et à mesure à l'aide d'un sous-problème de manière que, à chaque itération, uniquement les colonnes susceptibles de diminuer le coût total sont ajoutées à  $\Omega$ . La contrainte 2.1 traduit le fait que chaque ville est visitée par un seul véhicule et la contrainte 2.2 montre qu'on respecte le nombre de véhicules disponibles.

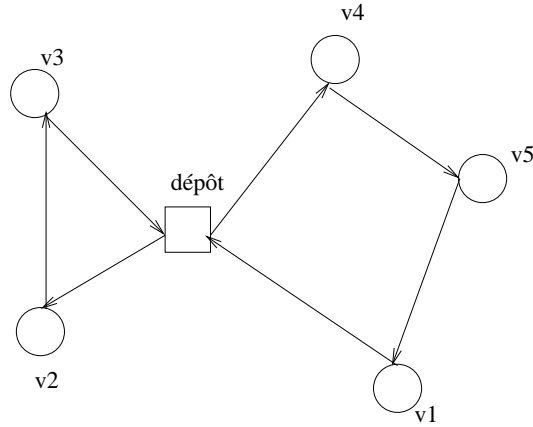


Figure 2.1 – Exemple de tournées de deux véhicules

$$C = \text{Min} \sum_{i=0}^m \sum_{j=0}^m \sum_{v=1}^n c_{ij} x_{ij}^v$$

sujet à :

$$\sum_{i=0}^m \sum_{v=1}^n x_{ij}^v = 1, \quad (j = 1..m) \quad (2.4)$$

$$\sum_{j=0}^m \sum_{v=1}^n x_{ij}^v = 1, \quad (i = 1..m) \quad (2.5)$$

$$\sum_{i=0}^m x_{ip}^v - \sum_{j=0}^m x_{pj}^v = 0, \quad (v = 1..n, p = 1..m) \quad (2.6)$$

$$\sum_{i=1}^m d_i \left( \sum_{j=0}^m x_{ij}^v \right) \leq K_v, \quad (v = 1..n) \quad (2.7)$$

$$\sum_{i=1}^m t_i^v \sum_{j=0}^m x_{ij}^v + \sum_{j=0}^m \sum_{v=1}^n t_{ij}^v x_{ij}^v \leq T_v, \quad (v = 1..n) \quad (2.8)$$

$$\sum_{j=1}^m x_{0j}^v \leq 1, \quad (v = 1..n) \quad (2.9)$$

$$\sum_{i=1}^m x_{i0}^v \leq 1, \quad (v = 1..n) \quad (2.10)$$

$$\sum_{v=1}^n \sum_{i \in R} \sum_{j \in R} x_{ij}^v \leq |R| - 1, \quad \forall R \subseteq \{1, \dots, m\} \quad (2.11)$$

$$x_{ij}^v \in \{0, 1\}, \quad \forall i, \forall j, \forall v \quad (2.12)$$

Après avoir présenté le modèle génération de colonnes, il s'avère important d'examiner un autre type de formulation. On note que  $m$  est le nombre de villes,  $n$  le nombre de véhicules disponibles,  $t_i^v$  représente le temps de service associé à la ville  $i$  et le véhicule  $v$ ,  $t_{ij}^v$  est le temps de déplacement associé au véhicule  $v$  entre les villes  $i$  et  $j$ , et finalement,  $K_v$  et  $T_v$  désignent respectivement la capacité et le temps maximum alloué à une tournée du véhicule  $v$ .  $x_{ij}^v$  est une véritable binaire qui vaut 1 si le véhicule  $v$  sert la ville  $i$  et  $j$  dans cet ordre (le dépôt 0 est considéré comme

une ville), sinon 0.

Les contraintes 2.4, 2.5 et 2.6 expriment le fait que chaque ville est desservie par exactement un véhicule, tout en imposant que le véhicule entrant à chaque ville soit celui sortant. Les contraintes 2.7 et 2.8 permettent de respecter respectivement la capacité et le temps maximal alloué à la tournée d'un véhicule. Les autres contraintes 2.9 et 2.10 garantissent qu'on ne dépasse pas le nombre de véhicules disponibles et, finalement, les contraintes 2.11 éliminent les sous-tours non connectés au dépôt.

Cette dernière formulation reste très peu utilisée pour les grandes instances, puisque le nombre de variables  $x_{ij}^v$  et le nombre de contraintes augmentent exponentiellement avec la taille du problème. Même si les contraintes 2.11 sont relaxées et ajoutées dynamiquement, on a quand même un nombre de contraintes de  $\Theta(mn)$  et un nombre de variables  $\Theta(nm^2)$ . Dans des cas pareils, une approche génération de colonnes, comme présentée dans le premier modèle, est généralement meilleure. Les variables (colonnes) ne sont pas toutes générées, de telle sorte que le nombre de variables reste raisonnable. En outre, les contraintes difficiles sont manipulées en général au niveau du sous-problème qui peut être résolu de manière exacte ou heuristique.

Certains travaux, comme ceux de Solomon [1987] et Solomon and Desrosiers [1988], ont introduit le problème de tournées de véhicules avec fenêtres de temps (PTVFT). Ce dernier dérive directement du PTV, en spécifiant que chaque visite de client (ville) doit respecter un intervalle de temps associé au client. Ainsi, si le client  $i$  est disponible uniquement pendant l'intervalle de temps  $[a_i, b_i]$ , alors sa visite doit être

planifiée au plus tôt au début de l'intervalle ( $a_i$ ) et au plus tard à la fin de l'intervalle ( $b_i$ ).

Le problème de chargement et déchargement avec fenêtres de temps (PCDFT) est une généralisation du problème du voyageur de commerce avec fenêtres de temps (PVCFT) ou le PTVFT dépendamment du nombre de véhicules disponibles. Cette généralisation consiste à spécifier que chaque demande (requête) est associée à la fois à un chargement et à un déchargement respectivement effectués dans un point de chargement et déchargement (voir Dumas et al. [2001], et Savelsbergh et Sol [1995]). En plus des contraintes usuelles qui définissent un PTVFT, le PCDFT doit respecter les contraintes de précedence permettant à chaque chargement de précéder son déchargement associé, tout en assurant que ces opérations soient effectuées par le même véhicule. Le PTVFT et le PCDFT se modélisent avec le modèle de partitionnement et se résolvent par génération de colonnes. Le premier met en oeuvre des sous problèmes de plus court chemin avec fenêtre de temps résolus par la programmation dynamique, alors que les sous problèmes du second nécessitent l'intégration des contraintes de précedence.

## 2.2 Transport en milieu forestier

Le problème du transport en contexte forestier (PTCF) présente des ressemblances avec certains problèmes de tournées de véhicules riches, notamment avec ceux du

PTV avec collecte et livraison. C'est que, dans le contexte forestier, les véhicules (camions) peuvent visiter plus d'une fois les points de chargement/déchargement (forêts/usines), étant donné que les demandes, en général, excèdent la capacité totale d'un camion. En plus, outre le fait que l'ordre est une décision importante dans une tournée de véhicule que ce soit pour le PTV ou le PTCTF, la quantité de bois chargée à partir d'une forêt vers une usine est une variable de décision que nous rencontrons aussi bien dans le PTCTF qu'au niveau du problème de transport avec collecte et livraison. Enfin, les deux problèmes considérés ont en commun la caractéristique consistant à avoir plusieurs points de chargement à partir desquels les livraisons sont assurées (forêts et dépôts).

Cette section portera essentiellement sur les principales contributions en transport forestier, en l'occurrence le transport des billes entre les sites de récoltes (forêts) et les usines. On passera en détails les différentes méthodes développées dans le cadre de la planification de l'horaire de camions (problème d'horaire, LTSP) transportant les billes de bois. On verra plus précisément des approches de planification d'horaire de camions conçues en avance et d'autres en temps réel pendant le transport, élaborées au fur et à mesure qu'on avance dans le temps jusqu'à la fin des tâches assignées. Pour clore, on abordera le concept de la synchronisation dans différents secteurs et contextes tels que les soins de santé à domicile, l'exploitation forestière et l'horaire d'équipages.



### 2.2.1 Planification tactique du transport de bois

Williamson et Nieuwenhuis [1993] ont proposé l'un des premiers travaux en optimisation du transport de bois. Ils ont mis en oeuvre un modèle mathématique permettant l'allocation optimale du bois entre les forêts et les usines. Leur approche s'inspire du travail de Papadimitriou et Steiglitz [1982] en déterminant le plus court chemin entre tous les points d'offre et de demande et en effectuant par la suite l'allocation des produits, suivant un ordre prédéfini. Carlsson et Rönnqvist [1998] et Palander et al. [2003] ont intégré dans leurs travaux l'allocation de flots de bois et les tournées des camions. Ces contributions ont permis essentiellement de concevoir de nouveaux tours dits complexes, au lieu de se contenter uniquement des tours simples (voir figure 2.2). Ces tours complexes sont constitués à partir de la fusion de tours simples. En pratique, seuls ceux formés en fusionnant deux tours simples ont été considérés. La conception d'un tour complexe répond à deux critères : premièrement, la distance parcourue à vide doit être plus petite que celle parcourue en tours simples ; deuxièmement, la distance totale du tour doit être inférieure à la distance maximale pouvant être effectuée par un camion durant un quart de travail. Au Canada, Gingras et al. [2007] ont développé une heuristique appelée MaxTour basée sur la méthode des gains de Clarke and Wright [1964]. À partir des demandes de transport établies au préalable par l'entreprise, l'heuristique propose des tournées combinant des voyages de manière à minimiser le temps parcouru à vide par les camions. Par conséquent, MaxTour permet une meilleure gestion, en réduisant les heures de travail improductives des camionneurs.

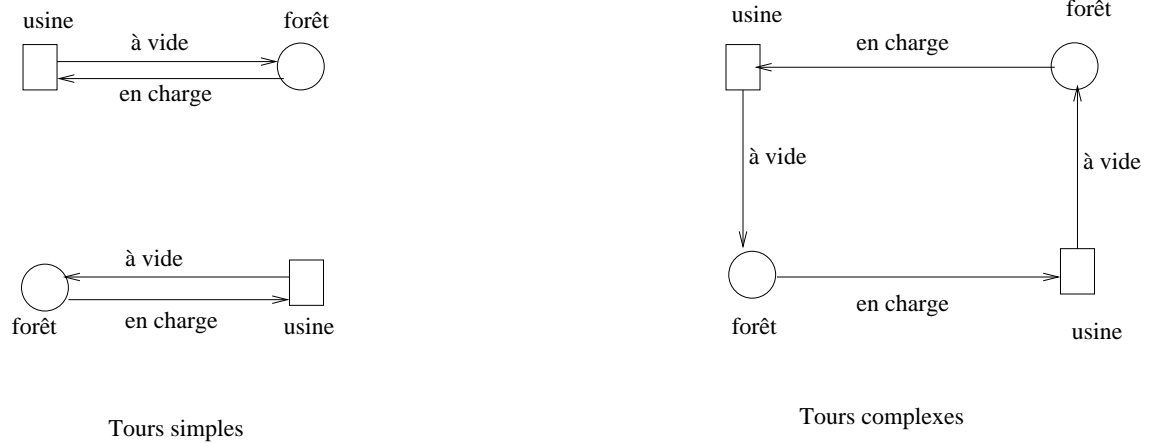


Figure 2.2 – Tours simples et complexes

### 2.2.2 Problème d’horaire du transport des billes de bois (LTSP)

La planification opérationnelle du transport forestier est divisée en deux étapes. La première est dite transport primaire. Elle consiste à transporter le bois abattu vers les bords des chemins. Au premier abord, l’abatteuse met des piles de bois à différents endroits de la zone de récolte. Ensuite, le débardeur collecte ces piles en formant à chaque voyage un chargement total en destination des bords des chemins. Récemment, Flisberg et Rönnqvist [2007] ont proposé un système supportant le transport primaire, réalisant jusqu’à 10% d’amélioration en gestion des routes des débardeurs et permettant une meilleure localisation du bois en forêts. La deuxième étape du transport est dite secondaire. Elle représente le transport du bois à partir des bords des chemins vers les usines ou les zones de stockage. Les camions remorques sont le moyen de transport le plus utilisé par rapport aux trains et aux bateaux. Traditionnellement, ce sont les contremaîtres (des spécialistes avec beaucoup d’expérience)

qui développent manuellement l'horaire et les routes des camionneurs. Ces dernières doivent respecter certaines contraintes, comme les quarts de travail des chauffeurs, les temps d'ouverture des usines et des forêts. On présente par la suite deux modèles classiques adaptés au LTSP qui diffèrent des modèles développés dans cette thèse. À la fin, il sera question, à partir de Palmgren [2005], de deux modèles classiques du LTSP dont on précisera avantages et inconvénients.

### Modèle non-linéaire de programmation en nombres entiers

Ce modèle est basé sur une formulation réseau qui contient trois types de variables (flot, temps et chargement). Le réseau associé repose sur deux ensembles principaux, l'un représentant les forêts ( $F$ ), l'autre les usines ( $U$ ). D'autres ensembles sont définis comme suite :

- $B^D$  : l'ensemble des noeuds associés aux bases de départ des véhicules
- $B^R$  : l'ensemble des noeuds associés aux bases de retour des véhicules allant de  $n + 1$  à  $n + nbV$ ,  $n$  étant le nombre de noeuds du réseau sans compter ceux de  $B^R$ ;  $nbV$  représente le nombre de véhicules disponibles
- $F_i$  : l'ensemble des copies du noeud  $i$  associé à la forêt  $i \in F$
- $A^F$  : l'ensemble de tous les noeuds associés à une forêt  $= \cup F_i$
- $U_i$  : l'ensemble des copies du noeud  $i$  associé à l'usine  $i \in U$
- $A^U$  : l'ensemble de tous les noeuds associés à une usine  $= \cup U_i$
- $N$  : l'ensemble formé à partir de tous les noeuds du réseau

Les arcs du réseau respectent certaines règles, ainsi tous les noeuds de  $B^D$  sont connectés à  $A^F$ . Les noeuds de  $A^F$  représentant un produit en forêts sont reliés à d'autres noeuds de  $A^F$  ayant le même produit en éliminant les arcs liant deux copies d'une même forêt; tout cela est dans le but de former des tournées de chargement

total d'un même produit. Tous les noeuds de  $A^F$  associés à un produit sont reliés à tous les noeuds de  $A^U$  associés au même produit. Les noeuds de  $A^U$  sont connectés à tous les noeuds de  $A^F$  et tous les noeuds de  $B^R$ . Finalement, pour permettre à un camion quelconque de rester dans sa base, on connecte son noeud associé de  $B^D$  à son noeud associé de  $B^R$ . Dans la suite, l'ensemble des arcs du réseau est noté par  $A$ . Les variables et notations suivantes sont utiles pour la suite du modèle.

|                                     |   |   |
|-------------------------------------|---|---|
| $x_{ijk}$                           | : | variable qui vaut 1, si le véhicule $k$ utilise l'arc $(i, j)$ , sinon 0          |
| $y_{ik}$                            | : | variable qui désigne la quantité chargée par le véhicule $k$ au noeud $i \in A^F$ |
| $q_{ijk}$                           | : | variable qui désigne la charge du véhicule $k$ en traversant l'arc $(i, j)$       |
| $s_{ik}$                            | : | variable représentant le temps d'arrivée du véhicule $k$ au noeud $i \in N$       |
| $V$                                 | : | l'ensemble de véhicules disponibles   |
| $t_{ij}$                            | : | le temps nécessaire pour traverser l'arc $(i, j)$                                 |
| $c_{ijk}(x_{ijk}, q_{ijk}, s_{ik})$ | : | le coût associé au véhicule $k$ pour traverser l'arc $(i, j)$                     |
| $[a_i, b_i]$                        | : | fenêtre de temps associée au noeud $i \in N$                                      |
| $p_k$                               | : | la capacité du véhicule $k \in V$   |
| $d_i$                               | : | la demande requise au noeud $i \in U$   |
| $r_i$                               | : | la quantité de bois disponible au noeud $i \in F$                                 |

Plusieurs contraintes sont implicitement gérées au niveau de la conception du réseau, comme le fait d'assurer que les produits sont correctement acheminés vers les usines et que chaque camion ne peut transporter qu'un seul type de produits par voyage, de façon à ce que, dans chaque voyage, une forêt, au moins, soit visitée. Les autres contraintes sont explicitement présentes dans le modèle :

$$\text{Min} \quad \sum_{(i,j) \in A} \sum_{k \in V} c_{ijk}(x_{ijk}, q_{ijk}, s_{ik})$$

sujet à :

$$\sum_{j:(i,j) \in A} x_{ijk} = 1, \quad \forall i \in B^D, \forall k \in V : i = k \quad (2.13)$$

$$\sum_{j:(i,j) \in A} x_{ijk} = 0, \quad \forall i \in B^D, \forall k \in V : i \neq k \quad (2.14)$$

$$\sum_{l:(l,i) \in A} x_{lik} - \sum_{j:(i,j) \in A} x_{ijk} = 0, \quad \forall i \in (N - B^R), \forall k \in V \quad (2.15)$$

$$\sum_{l:(l,i) \in A} x_{lik} = 1, \quad \forall i \in B^R, \forall k \in V : k = i - n \quad (2.16)$$

$$\sum_{l:(l,i) \in A} x_{lik} = 0, \quad \forall i \in B^R, \forall k \in V : k = i \neq n \quad (2.17)$$

$$q_{ijk} = 0, \quad \forall k \in V, \forall (i, j) \in A : i \in B^D \quad (2.18)$$

$$q_{ijk} = 0, \quad \forall k \in V, \forall (i, j) \in A : j \in B^R \quad (2.19)$$

$$\sum_{j:(i,j) \in A} q_{ijk} = 0, \quad \forall i \in A^U, \forall k \in V \quad (2.20)$$

$$x_{ijk} \left( \sum_{l:(l,i) \in A} q_{lik} + y_{ik} \right) = q_{ijk}, \quad \forall j \in N, \forall i \in A^F, \forall k \in V \quad (2.21)$$

$$q_{ijk} \leq p_k x_{ijk}, \quad \forall k \in V, \forall (i, j) \in A \quad (2.22)$$

$$\sum_{(i,j) \in A: j \in U_m} \sum_{k \in V} q_{ijk} \geq d_m, \quad \forall m \in U \quad (2.23)$$

$$\sum_{i \in F_j} \sum_{k \in V} y_{ik} \leq r_j, \quad \forall j \in F \quad (2.24)$$

$$s_{ik} = a_k, \quad \forall i \in B^D, \forall k \in V : i = k \quad (2.25)$$

$$x_{ijk}(s_{ik} + t_{ij}) \leq s_{jk}, \quad \forall (i, j) \in A, \forall k \in V \quad (2.26)$$

$$s_{ik} \in [a_i, b_i], \quad \forall i \in N, \forall k \in V \quad (2.27)$$

$$s_{ik}, y_{ik} \geq 0, \quad \forall i \in N, \forall k \in V \quad (2.28)$$

$$q_{ijk} \geq 0, \quad \forall (i, j) \in A, \forall k \in V \quad (2.29)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in V \quad (2.30)$$

Les contraintes 2.13 et 2.14 expriment le fait qu'un seul arc est utilisé pour le départ de chaque véhicule du noeud source. La contrainte 2.15 figure la conservation de flot à chaque noeud (sauf ceux de  $B^R$ ) et pour chaque véhicule. Les contraintes 2.16 et 2.17 traduisent le fait qu'un seul arc est utilisé pour le retour de chaque véhicule au noeud puit. Les contraintes 2.18 et 2.19 indiquent que chaque véhicule commence et finit vide à sa base. La contrainte 2.20 renvoie au fait que chaque véhicule n'a pas de chargement à faire en usine. La contrainte 2.21 permet de calculer la charge totale de chaque véhicule après chaque visite en forêt. La contrainte 2.22 montre qu'on respecte la capacité de chaque véhicule. Les contraintes 2.23 et 2.24 permettent d'assurer respectivement les demandes en usines et les stocks en forêts. Finalement, les trois contraintes restantes 2.25, 2.26 et 2.27 renvoient respectivement à l'heure de départ de chaque véhicule, à l'heure de visite de chaque noeud et aux fenêtres de temps associées aux forêts et aux usines. Ce modèle comporte un certain nombre d'inconvénients dont il faut signaler, en premier lieu, la fonction objectif et certaines contraintes qui sont non-linéaires ; ce qui complique la procédure de résolution et limite le choix des solveurs à utiliser. Le nombre de contraintes, de variables, de noeuds et d'arcs est non seulement lié au nombre de forêts et d'usines, mais aussi à la quantité de bois disponibles en forêts, à la quantité de bois demandée en usines et au nombre de véhicules disponibles. La structure du réseau est sensible à plusieurs règles qui diffèrent d'un contexte à l'autre et d'une compagnie à l'autre. Dans ce qui suit, on abordera un second modèle dont les variables représentent des routes réalisables pour un véhicule.

## Modèle de partitionnement avec contraintes additionnelles

Ce modèle implique principalement des variables binaires représentant les routes des véhicules, de sorte que chacune des routes respecte les contraintes de précédence, de capacité, de type de produits, de fenêtres de temps, . . . . Avant de décrire le modèle, on présentera les définitions nécessaires des variables et les notations utilisées.

|           |   |
|-----------|---|
| $n$       | : le nombre de véhicules disponibles  |
| $m_i$     | : le nombre de routes réalisables associées au véhicule $i$                                     |
| $n_f$     | : le nombre de forêts   |
| $n_u$     | : le nombre d'usines  |
| $s_f$     | : la quantité de bois disponible à la forêt $f$   |
| $d_u$     | : la demande associée à l'usine $u$   |
| $a_{ijk}$ | : la quantité de bois chargée à partir de la forêt $k$ par le véhicule $i$ suivant la route $j$ |
| $b_{ijp}$ | : la quantité de bois délivrée à l'usine $p$ par le véhicule $i$ suivant la route $j$           |
| $c_{ij}$  | : le coût de la route $j$ associé au véhicule $i$   |
| $p_k$     | : la pénalité associée au bois restant à la forêt $k$   |
| $g_p$     | : la pénalité associée à la demande non assurée de l'usine $p$                                  |
| $g'_p$    | : la pénalité associée au dépassement de la demande de l'usine $p$                              |
| $x_{ij}$  | : variable binaire qui vaut 1 si le véhicule $i$ suit la route $j$                              |
| $y_k$     | : variable associée à la quantité de bois en surplus à la forêt $k$                             |
| $u_p$     | : variable associée au déficit en demande de l'usine $p$  |
| $u'_p$    | : variable associée au surplus par rapport à la demande de l'usine $p$                          |

$$\begin{aligned} \text{Min} \quad & \sum_{i=1}^n \sum_{j=1}^{m_i} c_{ij} x_{ij} + \sum_{k=1}^{n_f} p_k y_k + \sum_{p=1}^{n_u} g_p u_p + \sum_{p=1}^{n_u} g'_p u'_p \\ \text{sujet à :} \quad & \sum_{j=1}^{m_i} x_{ij} = 1, \quad \forall i = 1, 2, \dots, n \end{aligned} \quad (2.31)$$

$$\sum_{i=1}^n \sum_{j=1}^{m_i} a_{ijk} x_{ij} + y_k = s_k, \quad \forall k = 1, 2, \dots, n_f \quad (2.32)$$

$$\sum_{i=1}^n \sum_{j=1}^{m_i} b_{ijp} x_{ij} + u_p - u'_p = d_p, \quad \forall p = 1, 2, \dots, n_u \quad (2.33)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \quad (2.34)$$

$$y_k, u_p, u'_p \geq 0, \quad \forall k, p \quad (2.35)$$

La contrainte 2.31 indique qu'on choisit, parmi toutes les routes réalisables, une seule route pour chaque véhicule. La contrainte 2.32 montre qu'on respecte le stock en forêt. Enfin, la contrainte 2.33 formalise la demande en usine. Le modèle en question permet de gérer la non réalisabilité du problème, lorsque le bois disponible en forêts ne suffit pas à couvrir toute la demande. Ce qui est extrêmement important ; car il est toujours utile d'avoir une solution partielle malgré le fait qu'il est impossible de satisfaire les demandes. Ce modèle de partitionnement avec contraintes additionnelles a un avantage crucial en comparaison avec le modèle précédent (non-linéaire), puisqu'il est indépendant des contraintes dues au contexte et aux règles des compagnies. Ces dernières sont gérées avec d'autres (précédence, capacité, etc) au niveau de la génération des routes. C'est ce qui donne beaucoup de flexibilité au modèle de partitionnement avec contraintes additionnelles.



## Approches résolvant le LTSP

Dans la littérature, les premiers à montrer un intérêt pour le LTSP furent Shen et Sessions [1989] qui ont développé un modèle de flux dans un réseau, minimisant les coûts de transport, tout en répondant à la demande et respectant les fenêtres de temps associées aux usines. Ce modèle a été résolu en utilisant l'algorithme «out-of-kilter» de Fulkerson [1961]. Dans leur travail, Weintraub et al. [1996] ont proposé un système informatisé appelé ASICAM, adapté au contexte chilien. Ce processus heuristique est basé sur quatre étapes que voici.

- Initialisation du temps  $t_0$  à 0.
- Evaluation de tous les voyages possibles entre  $t_0$  et  $t_0 + 60min$ .
- Assignment des voyages planifiés jusqu'à  $t_0 + 15min$ .
- S'il reste encore des clients (usines) ouverts, incrémenter  $t_0$  par  $15min$  puis revenir à l'étape 2, sinon arrêter.

L'évaluation des voyages est basée sur des règles heuristiques, et l'attribution des coûts aux voyages prend en considération la congestion au niveau des forêts. L'étape d'assignment de voyages repose sur des facteurs de priorités, comme une demande urgente faite par une usine, une demande en retard, .... ASICAM a permis une réduction de 32% en nombre de véhicules, de 13% en heures de travail et de coût, avec une augmentation de 31% en heures de productivité chez les compagnies.

Linnainmaa et al. [1995] ont mis au point le logiciel EPO utilisé largement par plusieurs compagnies finlandaises. Ce système traite les trois niveaux de planification, du stratégique à l'opérationnel, en se basant sur une approche combinant des méthodes heuristiques et exactes. Le niveau stratégique s'emploie à gérer les surfaces boisées à découper en prenant en compte les prévisions des demandes à satisfaire. Le niveau tactique traite du problème de l'allocation du bois, en adéquation avec les requêtes des usines. Le troisième niveau lui, se focalise sur la partie tournée des véhicules dont l'importance consiste à trouver des routes hebdomadaires optimales en terme de distance totale parcourue. Ce dernier niveau comprend trois phases. La première est réservée à la détermination des lots en partance vers les usines. La seconde est consacrée à l'élaboration de propositions relativement aux routes et aux horaires des camions. La troisième et dernière phase représente une post-procédure semi-manuelle où les conflits entre le plan découlant de la deuxième phase et les suggestions des contremaîtres se trouvent résolus. D'autres variantes du LTSP ont été traitées, au cours des dernières années, comme le cas spécial étudié par Jorgensen [1999] où chaque usine a un nombre limité de forêts à partir desquelles elle peut être approvisionnée. D'autres travaux, tel celui de Gronalt and Hirsch [2007], ont catégoriquement considéré que les requêtes de transport sont connues à l'avance. Gronalt and Hirsch [2007] ont appliqué une version modifiée de l'algorithme tabou unifié dont la taille des voisinages varie selon un mode oscillant.

Rönnqvist et Ryan [1995] ont proposé une méthodologie de résolution en temps réel en se basant sur le modèle de partitionnement avec contraintes additionnelles. Leur

approche se décompose en trois phases, les deux premières sont heuristiques, alors que la dernière représente une méthode exacte. L'étape heuristique initiale consiste à choisir les véhicules dans l'ordre où ils finiront leurs voyages en cours. Ensuite, de façon gloutonne, le meilleur prochain voyage de chaque véhicule est choisi. Une fois l'étape considérée terminée, ces voyages sont ajoutés à la liste utilisée par les planificateurs. De manière similaire, la deuxième étape heuristique est appliquée de façon séquentielle et gloutonne. Au lieu de se limiter à fixer le prochain voyage d'un véhicule, cette étape consiste à déterminer sa route entière. Le premier voyage de chaque route retenue est ajouté à la liste des planificateurs. Dans un contexte de génération de colonnes, ces routes trouvées constituent une base initiale pour résoudre un problème relaxé linéaire (PL). Une fois le PL résolu, la troisième étape, qui est une procédure de «branch and bound», est entamée, dans le but de trouver des solutions entières, en favorisant en premier lieu les véhicules prioritaires. La répartition de véhicules a été aussi utilisée par Rönnqvist et al. [1998] qui adoptèrent une approche en trois phases. La première énumère simplement tous les voyages ayant certaines caractéristiques. La deuxième permet de calculer le coût associé à chaque voyage. La troisième assigne les voyages aux véhicules disponibles, un par un, et ainsi de suite, en répétant la procédure jusqu'à la fin.

Palmgren et al. [2003] ont proposé une méthode basée sur le modèle générique. Leur approche génère a priori un ensemble de colonnes réalisables à partir de la solution optimale d'un problème de transport. Cet ensemble est ensuite utilisé pour résoudre la relaxation linéaire du problème maître (variables binaires considérées comme va-

riables continues non négatives), en appliquant le «pricing algorithm» qui consiste principalement à tarifier l'ensemble  $\Omega$  des colonnes générées a priori et à maintenir au fur et à mesure, parmi elles, un groupe déterminé comme actif. Pour obtenir des solutions entières, une approche «branch and price» permet à de nouvelles colonnes d'accéder à  $\Omega$  associé au problème restreint, suivie par la suite de la procédure «branch and bound». Dans le travail de Palmgren et al. [2004], les routes sont générées en résolvant un problème de  $k$ -plus courts chemins à chaque itération. La même procédure que celle de Palmgren et al. [2003] a été adoptée pour obtenir la meilleure solution entière possible. Finalement, dans sa thèse, Palmgren [2005] a décrit une méthode qui consiste à répéter plusieurs fois la phase de génération des routes a priori utilisée par Palmgren et al. [2003], avant de passer à la phase de recherche de la solution entière optimale.

Flisberg et al. [2009] et Andersson et al. [2008] ont mis en oeuvre une méthode de deux phases pouvant résoudre des instances réelles de très grandes tailles. La phase de début correspond à un problème de circulation recherchant les flux entre forêts et usines. C'est un simple problème de flot dans un réseau qui peut être formulé comme un LP. Une fois ce problème résolu, la construction des noeuds de transport est entamée en identifiant les ensembles possibles de chargement formant une charge totale de camion et en créant des noeuds de transport modélisant les changements de chauffeurs. En se basant sur ces noeuds de transport, un PTV est formulé dans le but de former les routes optimales des véhicules. La phase finale consiste à résoudre ce PTV via une extension de l'algorithme tabou unifié (voir Cordeau et al. [2001]).

## 2.3 Synchronisation

De nos jours, plusieurs applications exigent certaines formes de synchronisation. On cite, par exemple, les systèmes de distribution où les véhicules ne peuvent pas charger certains produits avant que d'autres véhicules ne les livrent. C'est ce qui arrive en forêts lorsque les camions doivent attendre le bois livré par les débardeurs pour le transporter des bords de chemin vers les usines. Dans les systèmes de transport en commun, les changements de chauffeurs de bus sont autorisés à des endroits bien précis. L'arrivée des bus à ces points suit un horaire prédéfini assurant implicitement la synchronisation d'arrivée des bus à ces endroits (voir Haase et al. [2001] et Freling et al. [2003]). Bredström and Rönnqvist [2008] proposent un modèle mathématique combinant la tournée et l'horaire de véhicules avec contraintes temporelles additionnelles, comme les contraintes de précédence ou de synchronisation. Pour illustrer l'importance de ces contraintes temporelles, les auteurs ont appliqué leur approche à différents domaines allant de la foresterie et de l'industrie aérienne aux soins de santé à domicile. Hane et al. [1997] décrivent une application liée à l'industrie aérienne qui consiste à affecter les types d'avions aux vols planifiés à l'avance, de sorte que les capacités associées aux types d'avions adhèrent aux prévisions de la demande. Ioachim et al. [1999] ont étudié une version modifiée de ce problème d'affectation, en assignant les avions aux vols suivant une base journalière sur un horizon hebdomadaire, tout en tolérant une flexibilité dans les horaires des départs. Ainsi, les contraintes de synchronisation indiquent que le même vol devrait débiter

à la même heure, chaque jour. Everborn et al. [2006] et Everborn et al. [2009] ont présenté un problème d’horaire des équipes de travail en soins de santé à domicile où certaines visites chez les clients (patients) doivent suivre un ordre prédéfini ou être simultanément réalisées dépendamment de la situation et des tâches attribuées aux équipes.

## CHAPITRE 3 : A HYBRID CONSTRAINT PROGRAMMING APPROACH TO THE LOG-TRUCK SCHEDULING PROBLEM

La programmation par contraintes englobe à la fois la modélisation et la résolution. Elle a été utilisée avec succès pour résoudre un grand nombre de problèmes combinatoires. Sa popularité est due principalement à son efficacité dans la résolution des problèmes d'ordonnancement et sa facilité à capturer certaines sous-structures globales des problèmes, contrairement à la programmation linéaire en nombres entiers où la modélisation des sous-structures d'un problème quelconque nécessite un effort supplémentaire.

Dans la littérature, plusieurs articles ont traité le problème de transport forestier avec horaire. Dans cet article, nous nous sommes intéressé particulièrement au problème journalier en tirant profit des avantages de la PPC et la PLNE. Ainsi, nous avons proposé une méthode hybride combinant la PLNE et la PPC de manière à ce que la PLNE modélise la circulation des camions par un modèle de flot à coût minimum. Cette solution optimale en termes de tournée de camion est communiquée par la suite à la PPC qui s'occupe de l'ordonnancement des activités associées au chargement et déchargement en tenant compte de contraintes de synchronisation entre les chargeuses et les camions pour minimiser les coûts d'attentes (grâce à la PPC ces coûts d'attentes sont facilement capturables). Outre la synchronisation, la contribution majeure de cet

article réside dans le fait que nous avons utilisé des contraintes globales de cardinalités dans la communication entre les deux modèles PLNE et PPC. À notre connaissance, ceci constitue une nouveauté dans les méthodes hybrides mettant en oeuvre la PPC et la PLNE. Par la suite, nous avons remarqué que la structure des routes que nous imposons à travers le modèle PLNE peut entraîner d'importants coûts d'attentes. Pour remédier à ce problème, nous avons introduit de la perturbation dans le modèle PLNE afin de diversifier les routes transmises au modèle PPC, dans le but d'avoir des routes quasi-optimales qui n'engendreront pas des coûts très élevés en temps d'attentes.

Une première partie de cet article a été publiée dans le compte-rendu de la conférence internationale *CPAIOR 2008* organisée à Paris. L'intégralité de l'article est en révision pour publication dans la revue internationale *Annals of Operations Research* en octobre 2009.



# A Hybrid Constraint Programming Approach to the Log-Truck Scheduling Problem

El hachemi Nizar,<sup>†</sup> Gendreau Michel,<sup>†</sup>  
and Louis-Martin Rousseau<sup>†</sup>

<sup>†</sup>*Interuniversity Research Centre on Enterprise Networks,  
Logistics and Transportation (CIRRELT)  
C.P. 6128, succursale centre-ville  
Montréal, Québec, Canada H3C 3J7*

{nizar, michelg, louism}@crt.umontreal.ca

## Abstract

Scheduling problems in the forest industry have received significant attention in the recent years and have contributed many challenging applications for optimization technologies. This paper proposes a solution method based on constraint programming and mathematical programming for a log-truck scheduling problem. The problem consists in scheduling the transportation of logs between forest areas and wood mills, as well as routing the fleet of vehicles to satisfy these transportation requests. The objective is to minimize the total cost of the non-productive activities such as the waiting time of trucks and forest log-loaders and the empty driven distance of vehicles. We propose a constraint programming model to address the combined scheduling and routing problem and an integer programming model to deal with the optimization of deadheads. Both of these models are combined through the exchange of global constraints. Finally the whole approach is validated on real industrial data.

**Keywords :** Forestry, transportation, routing, scheduling, constraint programming, hybrid method.

## 3.1 Introduction

The forest industry occupies an important place in the economy of several countries such as Chile, Canada, Sweden, Finland and New Zealand. Planning problems in forestry cover a wide scope of activities ranging from planting and harvesting to road building and transportation. Furthermore, in most problems, it is critical to pay attention to important environmental issues, as well as to company-specific goals and operating rules.

In Quebec, transportation represents more than 30% of the cost of provisioning for wood transformation mills, i.e., approximately \$15 per cubic meter of roundwood. Since the average distance between forest areas where wood is collected and mills to which this wood is transported is around 150 km, more than 50% of the fuel required per cubic meter of collected wood is consumed by the forest trucks traveling, half of the time empty, between forest areas and mills. It thus follows that transport activities between forest areas and mills should be organized as effectively as possible, both for economic and environmental reasons.

### 3.1.1 The Log-Truck Scheduling Problem

The Log-Truck Scheduling Problem (LTSP) is closely related to some routing problems encountered in other industries, in particular, so-called “pick-up and delivery problems” (see, for instance, Ropke et al. [2007]). In our case, we consider a pick-up and delivery problem in which for each request exactly one truckload of wood has to be transported from its pick-up location (forest area) to its delivery location (wood mill). A truck visits only one forest area and one mill on any given trip, i.e., requests are served individually by trucks. After unloading at a mill from its previous trip, a truck is usually sent back empty to its next forest destination, since all

requests are also assumed to be known in advance.

It must be noted that the problem as described above differs in several ways from the LTSPs addressed by some other authors. These differences stem from specific characteristics of Canadian forestry operations. Among others, in Canada, cut areas and log volumes are generally quite large. It is therefore customary when dealing with higher value products, such as hard wood, to assemble full truckloads prior to transportation by merging similar logs that are going to same destination. As a result, there is no need to explicitly take into account the several products (about 3 products in our case), since they are implicitly included in the full truckloads. While quantities are generally expressed in cubic meters, these can be easily converted into truckloads.

In this paper, we assume that we have predetermined destinations (a set of fixed transportation requests) as in Gronalt and Hirsch [2007]. In our case, the daily transportation requests are derived from decisions made at the weekly planning level.

In several variants of the LTSP, each truck must begin and end its route at its given home base, which often corresponds to the truck operator's home or yard. In our application, the bases of all trucks are the mills and we are allowed to reallocate trucks among the mills to obtain more efficient solutions. Driver changes are normally performed at mills when a route exceeds the maximum allowed driving time for a driver. In this paper, issues related to driving time, rests or changeovers are not taken into account ; these will be the subject of future work.

Furthermore, in each mill and each forest location, there is a single log loader that ensures the loading and unloading of all trucks. When a truck arrives at a location, if the loader is busy, then the truck has to wait until the loader becomes available. These waiting times can severely delay trucks and thus increase the cost

of transportation ; they should therefore be avoided as much as possible.

### 3.1.2 Literature Review

Since the mid-1990s, several companies in the forestry sector have initiated major projects aimed at improving the transportation portion of their activities, in particular, the control and quality of truck scheduling.

Weintraub et al. [1996] describe a heuristic-based model (ASICAM) that produces a daily plan for each truck by assigning loads and trips, while satisfying supply and demand constraints. The application of ASICAM in Chile has led to a 32% reduction in truck fleet size, a 13% reduction in average working hours and operational costs, and a 31% increase in productive hours for one company.

Another system called EPO developed by Linnainmaa et al. [1995] has been used in Finland. EPO is a system that deals with all stages from strategic to operative planning. The input data is collected on-line directly from the forest areas and the main output is a weekly schedule for each truck. A main goal of EPO is the minimization of truck driving as whole.

Palmgren et al. [2004] describe a near-exact method for solving the LTSP. This approach is based on column generation and pseudo-branch-and-price, where each column represents one feasible route for one truck. An initial set of routes is generated at the beginning. Then, the subproblem, which is a constrained shortest path problem, is solved by applying a k-shortest path algorithm. The columns whose reduced cost is negative will be communicated to the master problem.

Gronalt and Hirsch [2007] apply a modified version of the Unified Tabu Search heuristic to solve the LTSP where they vary the size of the neighborhoods in an oscillating fashion.

Flisberg et al. [2009] and Andersson et al. [2008] propose a two-phase solution approach that transforms the LTSP into a standard vehicle routing problem with time windows. The first phase determines the flow of wood from supply points to demand points, while the second phase combines transport nodes into routes. The dispatching procedure that continuously updates the trucks routes during the day is based on the previous work of Rönnqvist et Ryan [1995] and Rönnqvist et al. [1998].

With respect to numerical experiments, Murphy [2003] presents a case study with on average 9 trucks and 35 transport tasks per day, while Palmgren et al. [2003] have solve two case studies for Sweden : one with 6 trucks and 39 transport tasks, and one with 28 trucks and approximately 85 transport tasks. Gronalt and Hirsch [2007] solve random problems with 30 transport tasks and 10 trucks. Finally, Flisberg et al. [2009] and Andersson et al. [2008] have solved substantially larger instances ranging from 188 transport tasks to about 2,500 full truckloads with 15 to 110 trucks.

For a more detailed description of optimization problems in the forest sector, we refer the reader to Rönnqvist [2003].

There are many applications in which vehicles must be synchronized. This occurs in distribution systems when a vehicle cannot pick up a load until another vehicle has first delivered it, or in urban mass transit systems when drivers have to changebuses at so-called relief points (see Freling et al. [2003] and Haase et al. [2001]). Bredström and Rönnqvist [2008] present a mathematical programming model for the combined vehicle routing and scheduling problem with additional precedence and synchronization constraints. To illustrate the practical significance of the temporal precedence and synchronization constraints, the authors test their approach with the homecare staff scheduling problem and discuss the importance of synchronizaton constraints in the airline industry. They also discuss the need of coordination between harvesters

and forwarders in the forestry context. Since forwarding can only be done once the harvesting has been performed. With respect to the LTSP, the authors discuss the requirement to synchronize trucks (without crane) and loaders, since these loaders often serve several stands and move between these in order to load trucks.

### 3.1.3 A Constraint Programming Approach

Constraint Programming (CP) is a versatile paradigm for modeling and solving various practical combinatorial optimization problems. Over the last years, constraint-based scheduling has become an important tool for modeling and solving scheduling problems (see Baptiste et al. [1995, 2001]; Van Hentenryck [1989]; Zweben and Fox [1994]). In the context of the LTSP, it is a particularly attractive approach, because it can easily model both the synchronization of trucks and log loaders, and the optimization of the waiting costs.

Combining constraint programming and linear or mixed integer programming (MIP) has received a lot of attention since the late 1990's. One of the simplest approach consists in defining a subproblem that can be optimized through linear programming (LP) or MIP solver and whose optimal solution can be used by a CP master model (Sakkout et al. [2000]). Other forms of hybridization have been proposed for the Dantzig-Wolfe decomposition (Fahle and Sellmann [2000]; Sellmann et al. [2002]; Rousseau et al. [2002]); these take advantage of the flexibility of constraint programming to generate columns (decision variables) in a branch-and-price framework in the context of vehicle routing and crew rostering applications. Benders decomposition has been investigated by Hooker [2005] who combines MIP and CP to solve planning and scheduling problems : the MIP allocates tasks to facilities, while the scheduling is performed by CP ; the two are linked via logic-based Benders decomposition.

In a context similar to forest transportation (food industry transportation), Simonis et al. [2000] developed the TACT program whose solution approach consist of three steps : the first one determines the number of trips between farms and factories and the different truck types by solving an IP model; the second step takes these trips and schedules them in time; finally, the third step assigns individual resources to all activities. For more details about decomposition methods involving CP, we refer the reader to Milano [2004].

This paper presents a CP model for the LTSP based on an integer programming (IP) subproblem that first minimizes the total distance of deadhead trips. The solution of this IP is then used to generate global cardinality constraints for the CP scheduling model. To our knowledge this is the first time that the problem of synchronizing of trucks and log loaders is explicitly modeled and solved. Moreover, the contribution of this paper also lies in the communication between IP and CP through the use of structured global constraints, which we believe is novel.

The paper is organized as follows. Sections 3.2 and 3.3 present respectively the basic CP model and the solution approaches that we developed for solving the LTSP. The experimental setting is described in Section 3.4, where computational results are also reported. Section 3.5 concludes the paper.

## 3.2 Problem Description and Model

We now describe a CP model for the LTSP. CP presents many advantages in this context, such as allowing to easily express the difficult constraints (alternative assignment, routing) and to synchronize the trucks and the log loaders. The model is based on the scheduling language introduced by Van Hentenryck [1999] where problems are stated in terms of activities and ressources. Note that a unary resource is



a resource that cannot be shared by two activities (i.e., as soon as an activity requires the resource, no other activities can use of that resource). Alternative resources are sets of unary resources that are equivalent from the activity standpoint.

### 3.2.1 Parameters

|           |   |
|-----------|---|
| $D[m, f]$ | : distance between mill $m$ and forest area $f$ ( $= D[f, m]$ ),                |
| $nbR$     | : number of (real) transportation requests,                                     |
| $nbV$     | : number of vehicles,   |
| $R$       | : set of real transportation requests,  |
| $I$       | : $R$ augmented with dummy nodes representing the original locations of trucks, |
| $O$       | : $R$ augmented with dummy nodes representing the final destinations of trucks, |
| $F$       | : set of forest areas,  |
| $F_r$     | : forest origin of request $r$ ,  |
| $M$       | : set of wood mills,  |
| $M_r$     | : wood mill destination of request $r$ ,  |
| $V$       | : set of vehicles defined as <i>alternative unary resources</i> ,               |
| $L^m$     | : log loader at mill $m$ defined as a <i>unary resource</i> ,                   |
| $L^f$     | : log loader at forest area $f$ defined as a <i>unary resource</i> ,            |
| $T$       | : deadline for transporting all requests,                                       |
| $W$       | : domain of waiting time variables (0..T),                                      |
| $c_t$     | : cost of waiting one hour for a truck,   |
| $c_l$     | : cost of waiting one hour for a log-loader,                                    |
| $c_d$     | : hourly cost of driving an empty truck.  |

### 3.2.2 Variables and domains

The decision variables of the problem are based on the ILOG Scheduler component of OPL Studio 3.7 (see Van Hentenryck [1999]). For each *activity*  $A$ , two finite domain variables are created,  $A^s$  and  $A^e$ , which are associated respectively with the beginning and the end of the activity.

|                                |   |   |
|--------------------------------|---|---|
| $H$                            | : | <i>activity</i> of duration 0 that proceeds after all other activities, it constitutes the time horizon of the plan ; |
| $C_r, \forall r \in R$         | : | combined <i>activity</i> of loading, traveling and unloading request $r$ ;  |
| $P_r, \forall r \in R$         | : | pick-up <i>activity</i> of duration $d^p$ associated with request $r$ ;   |
| $D_r, \forall r \in R$         | : | delivery <i>activity</i> of duration $d^d$ associated with request $r$ ;  |
| $V_r \in V, \forall r \in R$   | : | vehicle assigned to request $r$ ;   |
| $S_r \in O, \forall r \in I$   | : | successor of request $r$ on the same vehicle ;  |
| $O_f, \forall f \in F$         | : | activity representing the opening time of forest area $f$ ;   |
| $W_r^f \in W, \forall r \in R$ | : | waiting time of a truck at the forest area of request $r$ ;   |
| $W_r^m \in W, \forall r \in R$ | : | waiting time of a truck at the wood mill of request $r$ ;   |
| $W_l^f \in W, \forall f \in F$ | : | waiting time of the log-loader in the forest area $f$ .   |

### 3.2.3 Constraints and Objective Function

Since we consider fixed transportation requests, several quantities are independent of the solution and can thus be removed from the objective function : these are the costs related to the loaded transportation distance as well as to the loading and unloading activities. Therefore it is sufficient to minimize the cost of non-productive activities, i.e. the waiting time of trucks and forest log-loaders and the empty driven distance of vehicles. Several reasons motivate this choice such as reducing the greenhouse gas emission and increasing the trucks productivity. In some variants of the LTSP studied in the literature, loaders work a fixed shift, and it is assumed that they support harvesting between truck loading operations. In our application, loaders cannot effectively support harvesting operations between successive trucks. as forest sites are too large. It is thus critical to maximise their usage.

$$\begin{aligned}
& \text{Minimize} && \sum_{r \in R} c_t(W_r^m + W_r^f) + \sum_{f \in F} c_l W_l^f + c_d \sum_{r \in R} D[M_r, F_{S_r}] \\
& \text{subject to} && \\
& C_r & \rightleftharpoons & V, \forall r \in R & (3.2.1) \\
& P_r & \rightleftharpoons & L^{F_r}, \forall r \in R & (3.2.2) \\
& D_r & \rightleftharpoons & L^{M_r}, \forall r \in R & (3.2.3) \\
& C_r^s & = & P_r^s, \forall r \in R & (3.2.4) \\
& C_r^e & = & D_r^e, \forall r \in R & (3.2.5) \\
& S_r & \neq & r, \forall r \in R & (3.2.6) \\
& S_{r_1} = r_2 & \Rightarrow & D_{r_1}^e + D[M_{r_1}, F_{r_2}] \leq P_{r_2}^s, \forall r_1 \in I, r_2 \in O & (3.2.7) \\
& V_{S_r} & = & V_r, \forall r \in I & (3.2.8) \\
& \text{ALLDIFFERENT}(S) & & & (3.2.9) \\
& \text{AHSR}(C_r, V, v) & \Leftrightarrow & V_r = v, \forall r \in I, \forall v \in V & (3.2.10) \\
& P_r^e + D[F_r, M_r] & = & W_r^m + D_r^s, \forall r \in R & (3.2.11) \\
& S_{r_1} = r_2 & \Rightarrow & W_{r_2}^f = P_{r_2}^s - D_{r_1}^e - D[M_{r_1}, F_{r_2}], \forall r_1, r_2 \in R & (3.2.12) \\
& O_f^e & = & \max(P_r^e | r \in R : F_r = f), \forall f \in F & (3.2.13) \\
& O_f^s & = & \min(P_r^s | r \in R : F_r = f), \forall f \in F & (3.2.14) \\
& W_l^f & = & O_f^e - O_f^s - \sum_{r \in R: F_r = f} d^p, \forall f \in F & (3.2.15) \\
& C_r & \prec & H, \forall r \in R & (3.2.16) \\
& H^e & \leq & T & (3.2.17)
\end{aligned}$$

Constraint (3.2.1) expresses the fact that each request *requires* ( $\rightleftharpoons$ ) a truck for its execution. Constraints (3.2.2) and (3.2.3) express the fact that loading and unloa-

ding *require* a log loader for their execution. Constraints (3.2.4) and (3.2.5) mean that the beginning time of a request must coincide with the beginning of its associated loading and that the ending time of a request coincides with the end of its associated unloading. Constraint (3.2.6) specifies that the successor of a request cannot be the request itself. Constraint (3.2.7) ensures that a truck that carries out two successive requests (including the dummy departures and ending requests) has enough time to do so. Constraint (3.2.8) expresses the fact that a request and its successor are served by the same truck. Constraint (3.2.9) ensures that all successors are given a different value (this, combined with the fact that a successor variable can only have one value, is equivalent to a flow conservation constraint). Constraint (3.2.10) is an OPL constraint (*ActivityHasSelectedResource*) that connects the transportation and scheduling component of the CP model (constraint *ActivityHasSelectedResource*( $a, S, u$ ) holds if activity  $a$  has selected resource  $u$  in the set of alternative resources  $S$ ). Constraints (3.2.12), (3.2.11) and (3.2.15) compute the waiting time of trucks at forest areas, wood mills and the forest log-loaders waiting time. Constraints (3.2.13) and (3.2.14) ensure that the activity  $O_f$  has a duration equal to the opening time of forest area  $f$ . Constraint (3.2.16) specifies that each request must be executed before the makespan dummy task ( $H$ ) (it precedes it ( $\prec$ )). The constraint (3.2.17) ensures that we respect the deadline to transport all the requests.

### 3.3 Solving the LTSP

We solved the proposed model using the Ilog *OPL-Studio* suite based on *Scheduler*, a specialized constraint programming library for modeling scheduling problem.

The search strategy consists of two steps. In the first step, we route the vehicles with the procedure *generate* of OPL Studio 3.7, which identifies a value for each

successor variable  $S_r, \forall r \in I$ . This instruction recursively identifies the variable which has the smallest domain and tries to assign it values in a lexicographic order. Since the values are ordered so that requests on any given forest site have consecutive numbers, this search strategy sends all trucks to the first site until all the requests have been serviced, which tends to minimize the opening time of a forest site and thus the unproductive time of its loader. We have implemented several other approaches, which aimed at reducing the deadheads or the waiting time of the trucks, but the solutions they produced showed an unacceptably low usage of forest loaders. In the second step, once the successor variables are fixed, we schedule each request as soon as possible by employing the built-in scheduling algorithms of Ilog Scheduler (*setTimes* and *rankGlobal*).

Finally, we explore the solution space using a dichotomic search procedure on the total cost and a depth-first search strategy (DFS) on all branching decisions (both for the routing and scheduling part). For a more detailed description of strategies, we refer the reader to Van Hentenryck et al. [2000].

### 3.3.1 Decomposing the LTSP

Since the cost of transportation is mainly based on distance, it is crucial to minimize the unproductive distance driven by empty vehicles. However, since the chosen search strategy does not attempt to reduce the traveled distance, we propose a decomposition approach where this criterion is explicitly addressed. We model the circulation of trucks between the mills and the forest areas as a network flow problem with some additional constraints that can be easily solved as an integer program (IP). This model yields an optimal solution with respect to the deadhead component of the objective function, but it is, however, not able to schedule the trucks and the log-

loaders. To link the IP and CP models, we have considered three different approaches :  
 1) Solve the IP, keep the optimal objective value, and use it as a constraint on the deadhead component of the objective function in the CP model ; 2) Solve the IP, keep the optimal solution, and use it to fix successor variables in the CP model ; 3) Solve the IP, look at the structure of the optimal solution, and impose it in the CP model through the introduction of global constraints on successor variables.

Since the objective function of the CP model is basically a large sum of small elements, it unfortunately does not allow for good back propagation (bounding the objective does not really trigger propagation and domain reduction). For that reason, method 1 would not be very useful in our context. On the other hand, since the IP model completely ignores the important scheduling aspects of the problem, fixing the sequence of all requests in the CP model would overly constrain the scheduler. Once the complete sequence is given, there is not enough flexibility left to avoid important waiting times. The challenge is thus to identify a good solution to the deadhead problem, while still giving the CP model enough flexibility to minimize waiting times in the final schedule.

For this reason we chose to migrate from the IP model only the minimal information that would allow achieving the minimal deadhead value. Since all full loads must be transported from their destination, we observed that the optimal value is completely determined by the arcs representing empty trips in the solution. It is thus not the global sequence that is important, but rather the number of empty trips performed between each mill and forest site.

These numbers can be extracted from the IP optimal solution and imposed in the CP model through the introduction of Global Cardinality Constraints (GCC, see Régim [1996]). We thus constrain the CP model to use the correct number of

deadhead trips between each mill-forest pair. Imposing this structure considerably reduces the search space and speeds up the resolution.

### 3.3.2 IP Model, variables, parameters and constraints

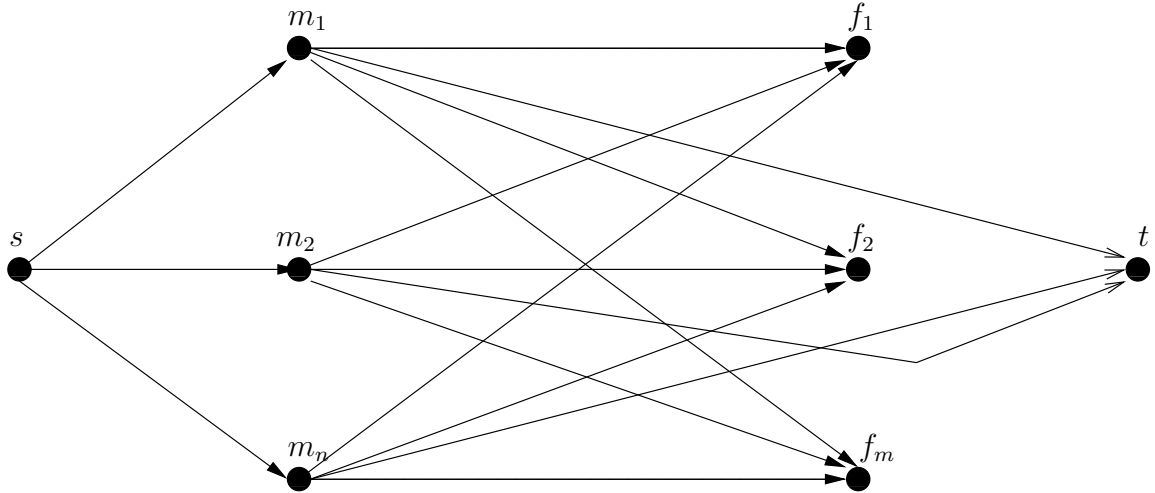


Figure 3.1 – Graph associated with the empty driving

In the Figure 3.1,  $s$  and  $t$  are two dummy nodes. The nodes on the left- and right-hand side represent respectively wood mills and forest areas. For each mill, we associate a demand that must be fulfilled, and each forest area provides a supply sufficient to meet the requests.

- $x_{ij}$  : number of empty trucks driving from woodmill  $i$  to forest area  $j$ ,
- $s_i$  : number of trucks that start at wood mill  $i$ ,
- $e_i$  : number of trucks that end up at wood mill  $i$ ,
- $d_i$  : number of requests associated with wood mill  $i$ ,
- $p_j$  : number of loads associated with forest area  $j$ .

The objective function of this IP model consists in minimizing the deadhead.

$$\text{Minimize } \sum_{i \in M} \sum_{j \in F} D[i, j] x_{ij} \quad (3.3.1)$$

$$\text{s.t. } \sum_{j \in F} x_{ij} + e_i = s_i + d_i, \quad \forall i \in M \quad (3.3.2)$$

$$\sum_{i \in M} x_{ij} = p_j, \quad \forall j \in F \quad (3.3.3)$$

$$\sum_{i \in M} s_i = |V| \quad (3.3.4)$$

$$\sum_{i \in M} e_i = |V| \quad (3.3.5)$$

$$x_{ij} \geq 0, \quad \forall i \in M, \forall j \in F \quad (3.3.6)$$

Let us  $(x_{ij}^*)$  be the optimal solution of the IP model. We define  $x_i^*$  as the vector composed of the  $|F|$  entries of  $x_{ij}^*$ . To introduce the GCC constraints, we need to define a new variable  $J_r^i$  which specifies which forest area will be visited just after unloading request  $r$  at mill  $i$ . The added variables and constraints are thus :

$$J_r^i = F_{S_r}, \quad \forall i \in M, \forall r \in I : M_r = i, \quad (3.3.7)$$

$$GCC(x_i^*, F, J_r^i), \quad \forall i \in M. \quad (3.3.8)$$

The starting and finishing wood mills associated to each truck are determined by the IP model, it is possible that a truck begins its day at a wood mill  $m$  and finishes it at an another one. For simplicity reasons, we use the linear programming solver (CPLEX) imbedded in OPL to solve the IP model.

### 3.3.3 Exploring other IP solutions

In some cases, it is possible that the flow structure of the IP optimal solution generates additional waiting time for trucks and forest log-loaders. It is thus interesting to explore other structurally different solutions of the IP model, which can be communicated to the CP model. These solutions could generate less trucks and



log-loaders waiting time than the optimal solution without deteriorating too much its value.

To generate new solutions, we perturb the objective function by introducing a penalty term on the value of the maximum flow in the IP solution. The model that we propose will no longer decrease the deadhead distance as the penalty coefficient increases (see Theorem 1). Basically, this means that we can increase the penalty coefficient until we consider that the empty driven distance is sufficiently deteriorated.

The objective function of this IP model is to minimize the deadhead plus penalty term which corresponds to the weighted value of the maximum flow. To present this model we need to define some new parameters and variables.

- $c$  : penalty coefficient.
- $v$  : the maximum number of empty trucks driving from a wood mill to a forest area,
- $P_c$  : the perturbed IP model associated with penalty coefficient  $c$ ,
- $X^*$  : a vector composed of the  $|F||M|$  entries  $x_{ij}^*$ ,
- $(X_c^*, v_c^*)$  : the optimal solution of  $P_c$ ,
- $W(X_c^*)$  : the empty driving distance of the optimal solution of  $P_c$ .

$$\text{Minimize } \sum_{i \in M} \sum_{j \in F} D[i, j]x_{ij} + cv \quad (3.3.9)$$

$$\text{s.t. } \sum_{j \in F} x_{ij} + e_i = b_i + d_i, \forall i \in M \quad (3.3.10)$$

$$\sum_{i \in M} x_{ij} = p_j, \forall j \in F \quad (3.3.11)$$

$$\sum_{i \in M} b_i = |V| \quad (3.3.12)$$

$$\sum_{i \in M} e_i = |V| \quad (3.3.13)$$

$$x_{ij} \leq v \quad \forall i \in M, \forall j \in F \quad (3.3.14)$$

$$x_{ij}, v \geq 0 \quad \forall i \in M, \forall j \in F \quad (3.3.15)$$

**Theorem 1** : Let  $c, c'$  be two integers such that  $c < c'$ , then  $W(X_c^*) \leq W(X_{c'}^*)$ .

**Proof**

Assume for the moment that  $v_c^* < v_{c'}^*$ .

Since  $(X_{c'}^*, v_{c'}^*)$  is the optimal solution associated with  $P_{c'}$ , we know that

$$W(X_{c'}^*) + c'v_{c'}^* \leq W(X_c^*) + c'v_c^*. \quad (3.3.16)$$

We have assumed that  $v_c^* < v_{c'}^*$  and we have  $c < c'$ , therefore :

$$(c - c')v_{c'}^* < (c - c')v_c^* \quad (3.3.17)$$

$$(3.3.16) + (3.3.17) \Rightarrow W(X_{c'}^*) + cv_{c'}^* < W(X_c^*) + cv_c^*, \quad (3.3.18)$$

which is impossible, since  $(X_c^*, v_c^*)$  is the optimal solution associated with  $P_c$ . It follows that

$$v_c^* \geq v_{c'}^*. \quad (3.3.19)$$

The fact that  $(X_c^*, v_c^*)$  is the optimal solution of  $P_c$  implies that

$$W(X_c^*) + cv_c^* \leq W(X_{c'}^*) + cv_{c'}^*. \quad (3.3.20)$$

$$(3.3.20) - c(3.3.19) \Rightarrow W(X_c^*) \leq W(X_{c'}^*) \quad (3.3.21)$$

As previously noted, the results of the  $P_c$  problem are communicated to the CP model by introducing new global cardinality constraints GCC.

### 3.4 Experimental Results

In this section we compare the three methods presented previously and evaluate their respective performance on two different case studies. These cases were provided by the Forest Engineering Research Institute of Canada (FPIInnovations<sup>1</sup>). The instances<sup>2</sup> come from two large timber companies. The first case involves six forest locations and five wood mills and the average cycle time to transport a load is 5.3 hours. The second case has five forest locations, five wood mills and an average cycle time of 4.7 hours. In both cases, loading and unloading times take around 20 minutes, with loading taking somewhat more time. With problem data discretized in five-minute units, the loading and unloading times of the first case are respectively 20 and 15. For the second case study the average loading time of 21 minutes and an average unloading of 18.5 minutes, have been both approximated to the nearest multiple of 5 minutes which is 20 minutes. The data of these case studies will be available on the web.

We ran several scenarios for each instance, varying the time horizon and amount of logs that needed to be transported (about 10-15 loads are carried every 6 hours). In the first set of experiments (reported in Tables 3.1 and 3.2), the number of vehicle varies between 14, 16 and 18 in order to ensure at least three and at most four trips per truck for each scenario. Each scenario was run for exactly 60 minutes using either the basic or the decomposition approach. The value of each solution is presented in

---

<sup>1</sup>The Forest Engineering Research Institute of Canada is a private, not-for-profit research and development organization whose goal is to improve Canadian forestry operations related to the harvesting and transportation of wood, and the growing of trees, within a framework of sustainable development.

<sup>2</sup>Instances are available at [www.crt.umontreal.ca/~louism](http://www.crt.umontreal.ca/~louism)

terms of unproductive costs. We report (in dollars) the deadhead costs, the waiting cost of trucks queueing to get loaded or unloaded, the waiting cost of log-loaders waiting for a truck to arrive, and finally the total cost of all these activities.

Table 3.1 – **Results of the first case study**

| $ V $                  | $ R $ | deadhead(\$) | truck(\$) | log-loaders(\$) | total(\$) |
|------------------------|-------|--------------|-----------|-----------------|-----------|
| Basic Approach         |       |              |           |                 |           |
| 14                     | 45    | 5600         | 40        | 2675            | 8315      |
| 16                     | 55    | 6907         | 80        | 2125            | 9112      |
| 18                     | 70    | 8680         | 1935      | 2442            | 13057     |
| Decomposition Approach |       |              |           |                 |           |
| 14                     | 45    | 5512         | 125       | 3633            | 9270      |
| 16                     | 55    | 6801         | 250       | 3675            | 10726     |
| 18                     | 70    | 8569         | 190       | 5608            | 14367     |

Table 3.2 – **Results of the second case study**

| $ V $                  | $ R $ | deadhead(\$) | truck(\$) | log-loaders(\$) | total(\$) |
|------------------------|-------|--------------|-----------|-----------------|-----------|
| Basic Approach         |       |              |           |                 |           |
| 14                     | 45    | 6399         | 95        | 5325            | 11819     |
| 16                     | 55    | 7845         | 170       | 5991            | 14006     |
| 18                     | 70    | 9800         | 325       | 6092            | 16217     |
| Decomposition Approach |       |              |           |                 |           |
| 14                     | 45    | 3202         | 145       | 4983            | 8330      |
| 16                     | 55    | 3914         | 25        | 5825            | 9764      |
| 18                     | 70    | 4982         | 45        | 8066            | 13093     |

Analyzing the results of Tables 3.1 and 3.2, we note that the cost of the unproductive activities increases proportionally with the size of the instances.

For the first case study, it seems that the decomposition could not improve the best solution of the straightforward method, as even a slight reduction of deadheads can significantly increase the waiting costs of the truck and the loaders. However, in the second case study, the decomposition method generally provides a better overall solution than the straightforward approach. As anticipated, for the larger instances, the log loader waiting time considerably increases as a result of the network structure that is imposed in the scheduling problem through the cardinality constraints.

When we look at the convergence of the two approaches (basic in Table 3.3 and decomposition in Table 3.4), we first notice that in almost half of the scenarios

the best solutions did not change after thirty minutes of computation time. We also can note that the decomposition approach converges more rapidly than the basic approach (half of the scenarios converged in 10 minutes by using the decomposition approach, whereas all the scenarios using the basic approach continued to be improved after 10 minutes). However, even when the decomposition approach continued to improve the solution after 10 minutes, the relative savings were not very important (less than 1% of the best solution). This behaviour is naturally explained by the fact that the solution space of the decomposed model is considerably smaller due to the added GCC constraints.

In order to improve the decomposition approach, we proposed the perturbed model (see subsection 3.3.3) that attempts to explore other circulations of trucks that have less impact on the log-loaders scheduling, but still have a very low deadhead cost. The result of this approach demonstrates that the structure imposed by the chosen vehicle circulation can have a significant impact on the objective function. Admitting the fact that the decomposition approach converges quickly, and that the possible benefit after 2 minutes of computational time is not considerable, we propose to vary the penalty coefficient as much as possible while respecting the total computational time. We limit the penalty coefficient to thirty ( $c \leq 30$ ), and for each value of penalty, we fix the computational time to 2 minutes. Thus the total computational time remains 60 minutes.

Table 3.3 – **Results of the first case study for six different computation times**

| $ V $                  | $ R $ | 2min  | 5min  | 10min | 15min | 30min | 60min |
|------------------------|-------|-------|-------|-------|-------|-------|-------|
| Basic Approach         |       |       |       |       |       |       |       |
| 14                     | 45    | 8488  | 8432  | 8432  | 8378  | 8315  | 8315  |
| 16                     | 55    | 9298  | 9194  | 9152  | 9152  | 9112  | 9112  |
| 18                     | 70    | 13182 | 13174 | 13165 | 13157 | 13132 | 13057 |
| Decomposition Approach |       |       |       |       |       |       |       |
| 14                     | 45    | 9304  | 9304  | 9270  | 9270  | 9270  | 9270  |
| 16                     | 55    | 10760 | 10760 | 10726 | 10726 | 10726 | 10726 |
| 18                     | 70    | 14434 | 14434 | 14421 | 14421 | 14407 | 14367 |

Table 3.4 – **Results of the second case study for six different computation times**

| $ V $                  | $ R $ | 2min  | 5min  | 10min | 15min | 30min | 60min |
|------------------------|-------|-------|-------|-------|-------|-------|-------|
| Basic Approach         |       |       |       |       |       |       |       |
| 14                     | 45    | 12146 | 12129 | 12054 | 11972 | 11972 | 11819 |
| 16                     | 55    | 14127 | 14118 | 14084 | 14068 | 14038 | 14006 |
| 18                     | 70    | 16251 | 16244 | 16225 | 16217 | 16217 | 16217 |
| Decomposition Approach |       |       |       |       |       |       |       |
| 14                     | 45    | 8402  | 8402  | 8345  | 8345  | 8345  | 8330  |
| 16                     | 55    | 9782  | 9782  | 9774  | 9774  | 9774  | 9764  |
| 18                     | 70    | 13098 | 13098 | 13093 | 13093 | 13093 | 13093 |



Table 3.5 – **Decomposition Approach with perturbation  $c \leq 30$** 

| $ V $                 | $ R $ | deadhead(\$) | truck(\$) | log-loaders(\$) | total(\$) |
|-----------------------|-------|--------------|-----------|-----------------|-----------|
| the first case study  |       |              |           |                 |           |
| 14                    | 45    | 5518         | 120       | 2350            | 7988      |
| 16                    | 55    | 6860         | 90        | 2483            | 9401      |
| 18                    | 70    | 8645         | 835       | 2217            | 11697     |
| the second case study |       |              |           |                 |           |
| 14                    | 45    | 3202         | 135       | 4700            | 8037      |
| 16                    | 55    | 3914         | 50        | 5750            | 9482      |
| 18                    | 70    | 4982         | 50        | 8066            | 13098     |

Looking at Tables 3.5, 3.1 and 3.2, we notice that in five of the six scenarios the perturbed decomposition provides a better or equivalent solution to the straightforward approach. Furthermore, on all our experiments, this later approach could always find its best solutions within 10 minutes of CPU time.

Interestingly, as we see in Figures 3.2 and 3.3, the main weakness of the decomposition approach with perturbation is that the perturbed IP model generates the same optimal solution for several penalty coefficients and we thus do not generate enough structurally different truck circulations to communicate to the CP model. Finding a technique to generate truck circulations that are significantly different is thus the subject of future research.

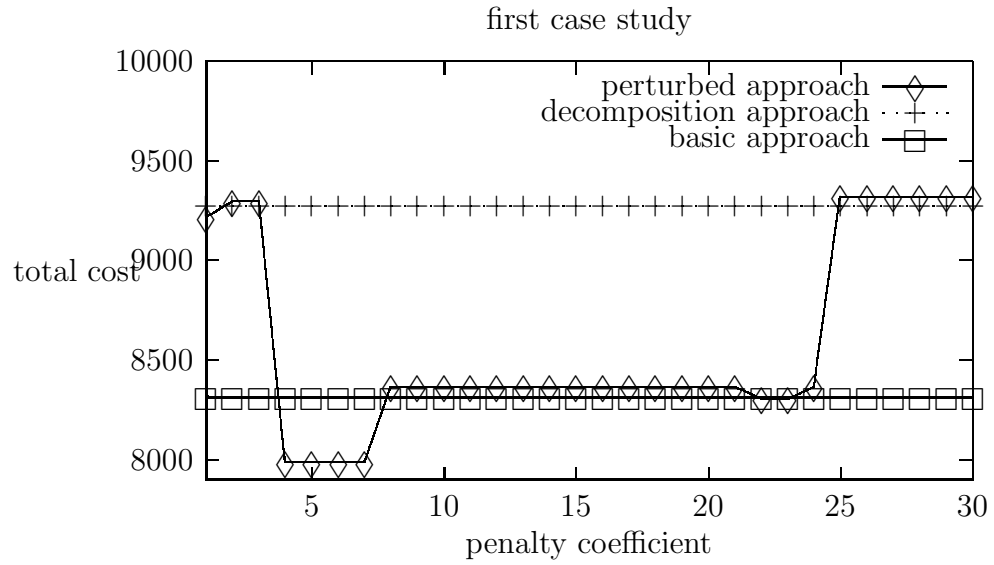


Figure 3.2 – Results of the first scenario of the first case study

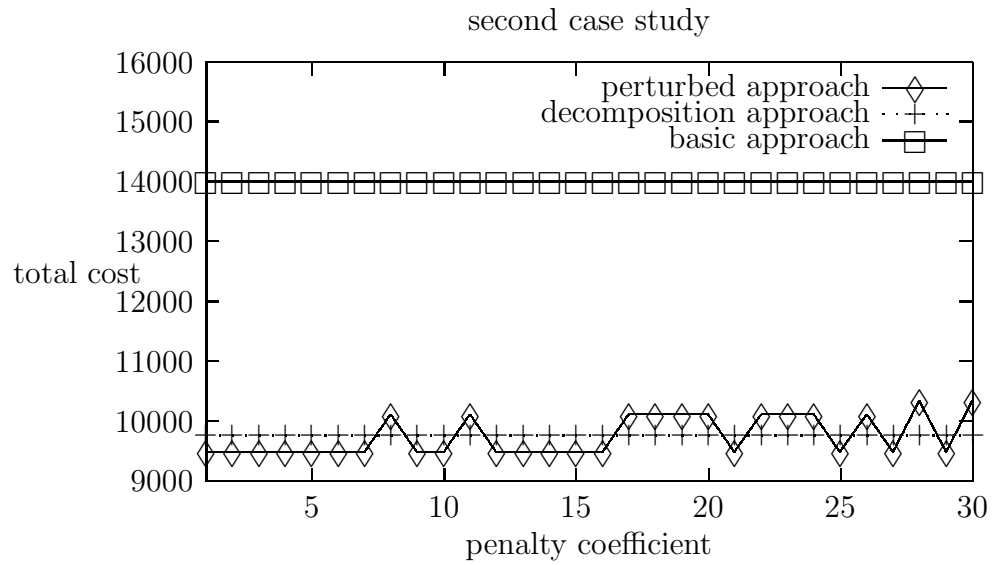


Figure 3.3 – Results of the second scenario of the second case study

### 3.5 Conclusion

We have presented a Log-Truck Scheduling Problem with synchronization constraint between the trucks and the log-loaders. To address this problem we proposed a decomposition approach based on Constraint Programming and Integer Programming models, which are combined through the communication of global constraints.

To avoid local minima, a perturbation technique was introduced in the IP model to explore different solutions that can be provided to the CP model. The whole process was then implemented through a multi-start strategy where we limit the CPU time in order to explore solutions generated with various values of the penalty term.

We believe that if we were able to generate and communicate structurally different optimal solutions of the IP to the CP model, we would be able improve the final result. Other research directions involve solving larger instances with a weekly long horizon and inventory constraints at the mill. These kinds of problems are of great interest to an industry who tries to adopt just-in-time delivery policies.

## CHAPITRE 4 : A HEURISTIC TO SOLVE THE SYNCHRONIZED LOG-TRUCK SCHEDULING PROBLEM

Si, comparativement aux autres méthodes (l'approche basique et l'approche décomposée), la décomposition avec perturbation nous a permis, au chapitre précédent, de trouver les meilleures solutions, elle a bien constitué notre source d'inspiration pour le présent chapitre quant à la génération de plusieurs circulation possibles, grâce à la recherche locale et à l'utilisation d'un algorithme glouton rapide qui nous a servi à élaborer les horaires de camions pour différents circuits.

La recherche locale basée sur les contraintes telle qu'elle est présentée par Van Hentenryck et Michel [2001] est un univers couplant les contraintes et la recherche locale. Ainsi, les contraintes sont utilisées pour guider la recherche vers les voisinages intéressants et deviennent partie prenante de la fonction objectif en évaluant le degré de la violation associé à chacune.

Dans cet article, nous proposons une heuristique pour résoudre un problème intégré de routage et ordonnancement avec contraintes de synchronisation relevant du domaine forestier. Nous nous sommes focalisé plus précisément sur le problème hebdomadaire de transport forestier avec horaire, en adoptant une politique de livraison en flux tirés juste à temps, permettant aux usines de recevoir leurs demandes journalières sans avoir à gérer des quantités très importantes en stocks. Nous avons développé une heuristique en deux phases. La première utilise un algorithme tabou dans le but de découper le problème en sept problèmes journaliers, de manière à minimiser le nombre de sites forestiers ouverts durant les jours de la semaine et les coûts de transport plein. La deuxième phase représente sept problèmes journaliers

de transport forestier avec horaire couvrant toute la semaine. Nous avons proposé un modèle RLBC intégrant la tournée et l'ordonnancement des camions, en tenant compte des contraintes de synchronisation entre les camions et les chargeuses, afin de minimiser les coûts d'attentes. Une recherche locale basée sur cinq structures différentes de voisinages gère la partie tournée, alors qu'un algorithme glouton s'occupe de l'ordonnancement à chaque itération après une modification des routes ; le tout est incorporé dans une structure d'algorithme appelée recherche locale itérative. Finalement, une deuxième approche est considérée dans cette seconde phase en introduisant un modèle PPC permettant une hybridation entre la RLBC et la PPC grâce à CO-MET1.1. Le rôle de cette hybridation est d'améliorer les résultats de l'algorithme glouton, surtout lorsque ce dernier prend de mauvaises décisions heuristiques.

Des parties de cet article sont publiées à la conférence internationale *MIC09* organisée à Hamburg en juillet 2009 et à *IESM09* organisée à Montréal en mai 2009. L'intégralité de l'article est soumise à la revue internationale *Computers and Operations Research* en octobre 2009.

# A Heuristic to Solve the Synchronized Log-Truck Scheduling Problem

El hachemi Nizar,<sup>†</sup> Gendreau Michel,<sup>†</sup>  
and Louis-Martin Rousseau<sup>†</sup>

<sup>†</sup>*Interuniversity Research Centre on Enterprise Networks,  
Logistics and Transportation (CIRRELT)  
C.P. 6128, succursale centre-ville  
Montréal, Québec, Canada H3C 3J7*

{nizar.el-hachemi, michel.gendreau, louis-martin.rousseau}@cirrelt.ca

## Abstract

We present in this article a synchronized routing and scheduling problem that arises in the forest industry, as a variation of the log-truck scheduling problem. It combines routing and scheduling of trucks with specific constraints related to the Canadian forestry context. This problem includes aspects such as pick-up and delivery, multiple products, inventory stock, multiple supply points and multiple demand points. We developed a decomposed approach to solve the weekly problem in two phases. In the first phase we use a tabu search algorithm to determine the destinations of full truckloads from forest areas to wood mills. At the second phase, we make use of two different methods to route and schedule the daily transportation of logs, the first one consists in using a constraint-based local search approach while the second one represents a hybrid approach involving a constraint programming based model and a constraint-based local search model. These approaches have been implemented using COMET1.1 which imbeddes a local solver, a MIP solver and a CP solver. We tested our method on a set of industrial cases from forest companies in Canada.

**Keywords :** Synchronized routing, scheduling, forestry, local search, constraint-based local search and constraint programming.

## 4.1 Introduction

Forestry represents a major economic sector in Canada and account for a large proportion of its exports. Planning problems in forestry cover a wide scope of activities ranging from planting and harvesting to road building and transportation. Furthermore, in many of these activities, it is critical to pay attention to important environmental issues, as well as to company-specific goals and operations rules, and government restrictions.

In Quebec, transportation represents more than 30% of the cost of provisioning for wood transformation mills, i.e., approximately \$15 per cubic meter of roundwood. The average distance between forest areas where wood is collected and mills to which this wood is transported is around 150 km, and about 50% of the fuel required per cubic meter of collected wood is consumed by the forest trucks traveling, half of the time empty, between forest areas and mills. Transport activities between forest areas and mills should therefore be organized as effectively as possible. If in recent years, significant attention has been devoted to transportation-related scheduling problems, mainly for economic and environmental reasons, in most Canadian forest companies, truck schedules are derived manually by an expert planner.

The main contributions of this paper are to synchronize trucks and log-loaders and to extend the LTSP to the weekly horizon, where inventories at wood mills are taken into consideration in the order to allow wood mills to work in a just in time mode. Furthermore, our approach is generally applicable to problems with a need to synchronize routing and scheduling.

The paper is organized as follows. Sections 4.2, 4.3, 4.4, 4.5 and 4.6 present respectively the literature review, the problem description, the solution approach, the tactical model, the daily Synchronized LTSP and the proposed algorithms. The



experimental setting is described in Section 4.7, where computational results are also reported. Section 4.8 concludes the paper.

## 4.2 Literature review

The log-truck scheduling problem (LTSP) is closely related to some routing problems encountered in other industries, in particular, so-called “pick-up and delivery problem.” For general surveys of the vehicle routing problem (VRP) and pick-up and delivery problem with time windows (PDPTW), we refer the reader to Cordeau et al. [2002], Gendreau et al. [2002], Savelsbergh et al. [1995], Dumas et al. [2001] and Toth and Vigo [2001]. In our case, we consider a pick-up and delivery problem in which for each request exactly one load of wood has to be transported from its pick-up location (forest area) to its delivery location (wood mill). A truck visits only one forest area and one mill on any given trip, i.e., requests are served individually by trucks. After unloading at a mill from its previous trip, a truck is usually sent back empty to its next forest destination.

Several models and methods have been developed in the literature to solve the LTSP. Among these, the heuristic-based approach of Weintraub et al. [1996] (ASICAM) has been used successfully since 1990 to produce daily plans for trucks in Chile. The application of ASICAM in Chile has led to a 32% reduction in truck fleet size, a 13% reduction in average working hours and operational costs, and a 31% increase in productive hours for one company. Linnainmaa et al. [1995] propose a knowledge-based system called EPO that deals with all stages from strategic to operational planning. The input data is collected on-line directly from the forest areas and the main output is a weekly schedule for each truck. One main goal of EPO is the minimization of truck driving as whole. Palmgren et al. [2004] have proposed more recently

a pseudo column-based model, which is solved using a branch-and-price procedure, where each column represents one feasible route for one truck. An initial set of routes is generated at the beginning. Then, the subproblem, which is a constrained shortest path problem, is solved by applying a k-shortest path algorithm. The columns whose reduced costs are negative are communicated to the master problem. Flisberg et al. [2009] and Andersson et al. [2008] proposed a two-phase solution approach, in which the LTSP is transformed into a standard vehicle routing problem with time windows. In the first phase, flows are determined from supply points to destinations, thus yielding a number of transportation “tasks” that are combined into routes in the second phase. The dispatching procedure that continuously updates the trucks routes during the day is based on the previous work of Rönnqvist et al. [1995] and Rönnqvist et al. [1998]. Gronalt and Hirsch [2007] applied a tabu search algorithm to solve a restricted variant of the LTSP in which the number of trips between each forest area and each mill is given. The same assumption is used by El Hachemi et al. [2008] who develop a hybrid method combining Integer Programming(IP) and Constraint Programming(CP) : the IP model generates optimal routes in term of deadheading, while CP deals with the scheduling part. For a more detailed description of optimization problems in the forest sector, we refer the reader to Rönnqvist [2003].

With respect to numerical experiments, Murphy [2003] presents a case study with on average 9 trucks and 35 transport tasks per day, while Palmgren et al. [2003] have solved two case studies for Sweden : one with 6 trucks and 39 transport tasks, and one with 28 trucks and approximately 85 transport tasks. Gronalt and Hirsch [2007] solved random problems with 30 transport tasks and 10 trucks, El Hachemi et al. [2008] deal with daily problems having up to 18 trucks and 70 transport tasks. Finally, Flisberg et al. [2009] and Andersson et al. [2008] have solved substantially

larger instances ranging from 188 transport tasks to about 2.500 full truck loads with 15 to 110 trucks.

Synchronizing vehicles with others tasks arises in many real-world applications. The synchronization constraints occur in the well known dial-a-ride problem (see Rousseau et al. [2003]), in which some disabled persons require assistance in order to prepare for transportation. This assistance can vary from dressing to providing a wheelchair and is not usually required at all time or by all customers. More details concerning the homecare staff scheduling are given by Everborn et al. [2006] and Everborn et al. [2009]. There are many other applications in which vehicles must be synchronized, like the urban mass transit systems, which combines vehicle and crew scheduling problems. The drivers are allowed to change bus in so called relief points. As described in Haase et al. [2001] and Freling et al. [2003], the synchronized arrival of bus to the relief points is implicitly guaranteed, since it is defined by a timetable. In the airline industry, synchronization constraints have been introduced by Ioachim et al. [1999]. In fact, the classical fleet assignment problem studied by Hane et al. [1997] states that, given a fixed flight the timetable, there is a need to decide which aircraft to use for each flight on a daily basis. Of course, the capacity of each aircraft type must match with the expected demand of passengers on all flights. However, Ioachim et al. [1999] studied the case where timetable is not fixed and departure times of flights are given on a daily basis for an entire week are within time windows. To obtain a more robust schedule, synchronization constraints are introduced to guarantee that every flight will have the same departure time on each day. In the forest industry, harvesting operations are performed by harvesters and forwarders. The coordination between them specifies that forwarding can only be done once harvesting has been performed (see Karlsson et al. [2004]). In the same context, and related to the LTSP, there is a need to synchronize trucks those that have not an onboard crane

and loaders, since these loaders often serve several stands and move between them in order to load trucks. For more details about the importance of synchronization constraints in the real-world situations, we advise the reader to see Bredström and Rönnqvist [2008].

### 4.3 Problem description

It must be noted that the problem as considered in this paper differs in several ways from LTSPs addressed by other authors. These differences stem from specific characteristics of Canadian forestry operations. Among others, in Canada, cut areas and log volumes are generally quite large. It is therefore customary when dealing with higher value products, such as hard wood, to assemble full truckloads prior to transportation by merging similar logs that are going to same destination. While quantities are generally expressed in cubic meters, these can be easily converted into truckloads. In several variants of the LTSP, each truck must begin and end its route at its given home base, which often corresponds to the truck operator's home or yard. In our application, the bases of all trucks are mills and we are allowed to reallocate trucks among mills to obtain more efficient solutions. Driver changes are normally performed at mills when a route exceeds the maximum allowed driving time for a driver. In this paper, issues related to driving time, rests or changeovers are not taken into account. Demands at wood mills are given on a daily basis, whereas routes and schedules of trucks are determined on a weekly basis. Since the stock of products at wood mills are constrained, this requires integrating transportation schedules over several days.

We assume that at each mill and each forest location, there is a single log loader that ensures the loading and unloading of all trucks. When a truck arrives at a loca-

tion, if the loader is busy, the truck has to wait until the loader becomes available. These waiting times can severely delay trucks and thus increase the cost of transportation; they should therefore be avoided as much as possible. In some variants of the LTSP studied in the literature, loaders work a fixed shift during the day, and it is assumed that they support harvesting between truck loading operations. In our application, loaders cannot effectively support harvesting operations between successive trucks as forest sites are too large. It is thus critical to maximize their usage. To reduce the greenhouse gas emission and increase the trucks productivity, loaded and unloaded travel distance cost have been taken into account in our approach.

## 4.4 Solution Approach

This paper presents novel two-phase methods for a variant of the weekly LTSP integrating synchronization constraints between trucks and loaders. We propose a MIP model called “tactical model” to decompose the weekly problem in seven daily problems that minimizes the number of forest areas operating days of the week combined with the loaded travel cost. To address the daily LTSP and deal with synchronization constraints, we first develop a straightforward approach based on constraint-based local search (CBLS) and the COMET1.1 programming language. A second approach follows directly from the first one by hybridizing the CBLS model with a previously CP model proposed for LTSP (see El Hachemi et al. [2008]) in some specific situations.

CBLS consists in using constraints and objective function to describe and control the local search. The invariants (incremental variables), or one-way constraints provide declarative specifications of incremental algorithms, they have been introduced by Michel and Van Hentenryck [1997] and Michel and Van Hentenryck [2000]. More

precisely, they specify what to maintain incrementally. They are useful for maintaining various neighborhoods incrementally and evaluate the impact of complex moves on differentiable objects (objective function). The project COMET was initiated in 2001 to explore the CBLs potentiality and how it could be supported in high-level programming languages (see Van Hentenryck et Michel [2001]). Actually, COMET supports both modeling and search abstractions in the spirit of constraint programming and includes CP, CBLs and linear solvers.

CP is a versatile paradigm for modeling and solving various practical combinatorial optimization problems. Over the last years, constraint-based scheduling has become an important tool for modeling and solving scheduling problems (see Baptiste et al. [1995], Baptiste et al. [2001], Van Hentenryck [1989] and Zweben and Fox [1994]). In the context of the synchronized LTSP, it is a particularly attractive approach, because it can easily model the synchronization of trucks and log-loaders and the optimization of the waiting costs. CP represents loading and unloading times as domain variables and by propagating the synchronization constraints only when needed, it allows an efficient implementation of those constraints.

## 4.5 The Tactical Model

We present in this section our tactical planning model and its associated tabu search solution procedure. In this model, we must take into consideration the fact that different wood products must be shipped, because logs are sorted into different assortments. The assortment type depends on species, usage, quality, and dimension. Many wood mills order logs in specific lengths and diameters to produce given final products.

### 4.5.1 Parameters

|             |  |
|-------------|--|
| $F$         | : The set of forest areas.   |
| $W$         | : The set of wood mills.   |
| $P$         | : The set of forest products.  |
| $J$         | : The set of days of the week.   |
| $S_p^{max}$ | : The daily maximum number of full truckloads of product $p$ that can be stocked at a wood mill.                         |
| $U$         | : The daily upper bound on the number of full truckloads that can be transported from any supply point.                  |
| $L$         | : The daily lower bound on the number of full truckloads that must be transported from any supply point if it is opened. |
| $D_{wp}^j$  | : The demand of product $p$ at wood mill $w$ on day $j$ .  |
| $C^*$       | : The fixed cost of opening any forest area for a day.   |
| $c_{fw}$    | : The travelling cost between forest area $f$ and wood mill $w$ .  |

### 4.5.2 Variables of the Model

|                |   |
|----------------|---|
| $open_f^j$     | : A binary variable equal to 1 if forest area $f$ is opened day $j$ .   |
| $trip_{fwp}^j$ | : An integer variable representing the number of full truckloads of product $p$ from supply point $f$ to wood mill $w$ on day $j$ . |
| $stock_{wp}^j$ | : An integer variable representing the number of full truckloads of product $p$ stocked at wood mill $w$ on day $j$ .               |

### 4.5.3 Constraints and Objective

$$\text{Minimize } \sum_{j \in J} \sum_{f \in F} C^* open_f^j + \sum_{j \in J} \sum_{f \in F} \sum_{w \in W} \sum_{p \in P} c_{fw} trip_{fwp}^j$$

subject to

$$stock_{wp}^j \leq S_p^{max}, \forall w \in W, \forall p \in P, \forall j \in J \quad (4.5.1)$$

$$stock_{wp}^{j-1} + \sum_{f \in F} trip_{fwp}^j = D_{wp}^j + stock_{wp}^j, \forall w \in W, \forall p \in P, \forall j \in J \quad (4.5.2)$$

$$\sum_{w \in W} \sum_{p \in P} trip_{fwp}^j \leq U \cdot open_f^j, \forall f \in F, \forall j \in J \quad (4.5.3)$$

$$\sum_{w \in W} \sum_{p \in P} trip_{fwp}^j \geq L \cdot open_f^j, \forall f \in F, \forall j \in J \quad (4.5.4)$$

In this phase, the objective function is to minimize the sum of opened sites cost and the loaded travel cost. We decided to set  $C^* \gg c_{fw}$ . This choice is motivated by the fact that the optimizer will then seek to minimize the number of sites in operation during the week. This will induce global control over log-loaders waiting time cost on each day, since no costs are attributed in this model to inactive loader during the whole day.

Constraint (4.5.1) ensures that the daily stock of any product at any wood mill respects the maximum limit fixed. Constraint (4.5.2) specifies that demand and stock conservation is satisfied over the whole week. Constraint (4.5.3) expresses the fact that each loader has a finite capacity to load logs (a loader cannot exceed a number of loads per day). Finally, constraint (4.5.4) indicates that if a loader is operating during one day, it must serve a minimum number of trips.

#### 4.5.4 Solution Method

We have developed a local improvement algorithm with a tabu component to solve the tactical model. This algorithm is divided into two stages executed sequentially.

First, we generate an initial solution by assuming that all forest areas are opened all week [see figure 4.1 representing the network associated with two forest areas and two wood mills and only one product type], and applying a feasible flow algorithm. This is possible because the tactical problem becomes a simple minimum cost flow problem once the binary variables  $open_f^j$  are fixed. We apply a feasible flow algorithm instead of a minimum cost flow algorithm because the objective of this first stage is simply to find a feasible solution with the minimum number of forest areas opened. We denote by  $s$  a solution of the tactical model. We can represent  $s$  only by its



binary variables  $open_f^j$ ,  $s = \langle open_1^1, \dots, open_{|F|}^{|J|} \rangle$ . The neighborhood used during the first part is simple : it consist in changing the status of any opened forest area on any day to the closed status.  $N_1(s) = \{flip(s, open_f^j) \mid open_f^j = true\}$ . We have two one-dimentional array tabu list, one associated with the forest areas and the other with the days. If the binary variable  $open_f^j$  is related to a move  $s^* \rightarrow s$ , we impose that forest area  $f$  and day  $j$  be tabu for a few iterations thus preventing during that period the closure of forest area  $f$  on any other day, as well as the closure of any other forest area on day  $j$ .

After the completion of the first part (see figure 4.2), the second part is executed with the goal of minimizing the loaded travel cost. During this step, the number of forest areas opened during the week remains constant. This is enforced by the structure of neighborhood used,  $N_2$ . This neighbourhood includes for every day  $j$  all pairs of forest areas with different values.

Formally,  $N_2(s) = \{flip(s, open_{f_1}^j, open_{f_2}^j) \mid open_{f_1}^j \neq open_{f_2}^j\}$ . Besides the neighborhoods, the second part differs from the first one in four important ways : 1) We accept moves to solutions that deteriorate the objective function ; 2) We use a minimum cost flow algorithm instead of a feasible flow algorithm to determine the flows associated with a solution  $s$  ; 3) The objective function now represents the loaded travel cost ; 4) Tabus are now stored in a matrix format and prohibit changing again the status of a forest area on a given day for some iterations. However, we keep the one-dimensional array tabu list associated with the days to guide local search through various days in consecutive iterations (see figure 4.3). We fixed a limit  $\epsilon$  to the deterioration ( $\delta$ ) of the objective function, this limit can be changed dynamically during this stage.

Finally, at the output of this step, seven input data associated with each day of the week representing the requests to transport from forest areas to wood mills will

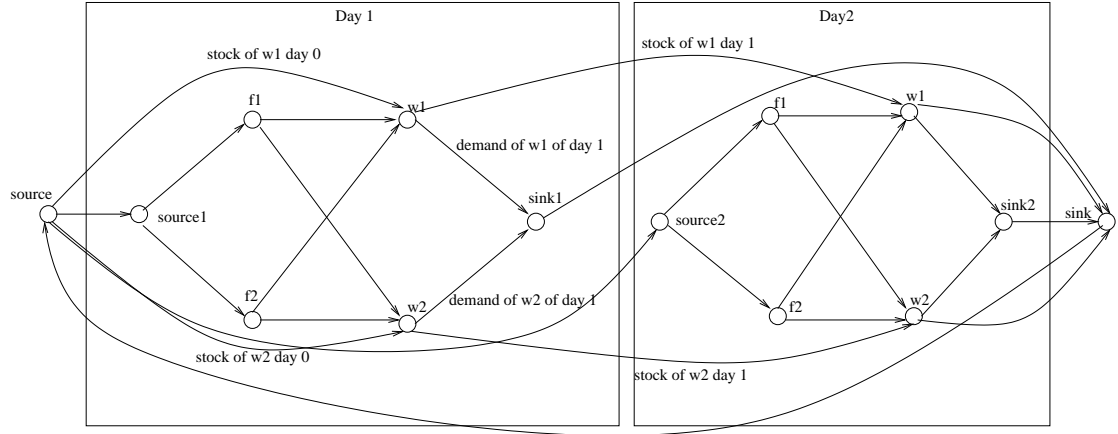


Figure 4.1 – Network example of tactical model

be communicated to the different daily log-truck scheduling problems.

It should be noted that we tried to solve the tactical model using the COMET MIP Solver (lp solve5.5), but failed to obtain any feasible solution within an hour of computation. A possible cause for this failure might be that the parameters of this MIP Solver cannot be adjusted for the time being (COMET MIP Solver is used as a black box).

## 4.6 The Daily Synchronized Log-Truck Scheduling Problem

To deal with the daily synchronized LTSP, we developed two approaches. The first one involves a CBLSP model solved with an iterated local search algorithm (ILS) integrating a vehicle routing algorithm based on various neighborhoods and a greedy algorithm to schedule the transportation of logs once routes are fixed. The second

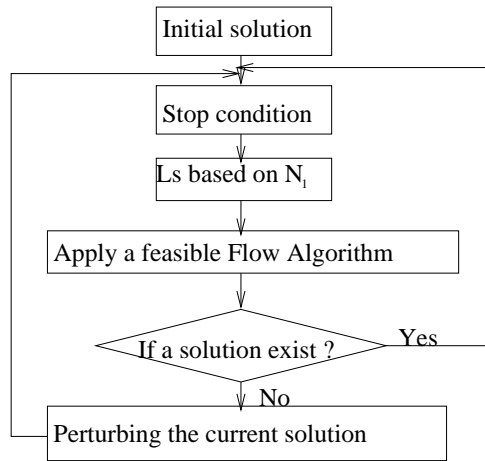


Figure 4.2 – First part of the tactical problem algorithm

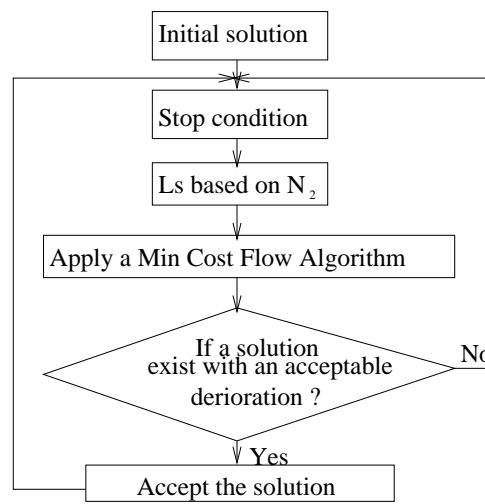


Figure 4.3 – Second part of the tactical problem algorithm

approach consists in hybridizing the previous CBLS model with a CP model. In this approach we keep the same routing algorithm, however, for the scheduling part, we use the greedy algorithm for the CBLS model and the COMET CP solver when dealing with the CP model. Since CP can be relatively time consuming, we use it only occasionally. The best solution identified during the execution of the algorithm is then used as an upper bound in the CP model, forcing it to improve the quality of the best solution found so far.

#### **4.6.1 A Constraint-Based Local Search Model**

We now describe all components of our ILS algorithm. Thus we will present the CBLS model, the routing algorithm and its initial solution, and finally the greedy algorithm.

#### 4.6.1.1 Parameters

|           |  |
|-----------|--|
| $D[m, f]$ | : The travel time between a mill $m$ and a forest area $f$ .             |
| $nbR$     | : The number of real transportation requests.                            |
| $nbV$     | : The number of vehicles.  |
| $R$       | : The set of real transportation requests $[1..nbR]$ .                   |
| $F_r$     | : The forest origin of request $r$ .                                     |
| $M_r$     | : The wood mill destination of request $r$ .                             |
| $I$       | : $R$ added with departures from initiale locations $[(1 - nbV)..nbR]$ . |
| $O$       | : $R$ added with empty returns to final locations $[1..(nbR+nbV)]$ .     |
| $T$       | : The deadline to tranport all requests.                                 |
| $c_t$     | : The cost of waiting time of a truck per unit of time.                  |
| $c_l$     | : The cost of waiting time of a log-loader per unit of time.             |
| $c_d$     | : The cost of an empty driving per unit of time.                         |
| $d_p$     | : The duration of a pick-up activity at a forest area.                   |
| $d_d$     | : The duration of a delivery activity at a wood mill.                    |
| $L^f$     | : A log-loader assigned to forest area $f$ defined as a unary resource.  |
| $L^m$     | : A log-loader assigned to wood mill $m$ defined as a unary resource.    |
| $W$       | : The domain of waiting time variables $[0..T]$ .                        |

#### 4.6.1.2 CBLS Model Variables

|               |  |
|---------------|--|
| $Start_{P_r}$ | : The beginning time of the pick-up activity related to request $r$ .    |
| $Start_{D_r}$ | : The beginning time of the delivery activity related to request $r$ .   |
| $Prec_r$      | : The request preceding the request $r$ on their associated truck route. |

#### 4.6.1.3 CBLS Model Invariants

- $End_{P_r}$  : The ending time of the pick-up activity related to request  $r$ .
- $End_{D_r}$  : The ending time of the delivery activity related to request  $r$ .
- $Wait_{M_r}$  : The waiting time of a truck at a wood mill related to request  $r$ .
- $Wait_{F_r}$  : The waiting time of a truck at a forest area related to request  $r$ .
- $Start_{F_f}$  : The starting time of work of the log-loader associated with forest area  $f$ .
- $End_{F_f}$  : The ending time of work of the log-loader associated with forest area  $f$ .

#### 4.6.1.4 CBLS Model and Formulation of the Synchronized LTSP

We describe our log-truck scheduling model with the objective function to minimize the cost of unproductive activities.

$$\text{Min } \sum_{r \in R} c_d D[M_{Prec_r}, F_r] + \sum_{r \in R} c_t (Wait_{M_r} + Wait_{F_r}) + \sum_{f \in F} c_l (End_{F_f} - Start_{F_f})$$

subject to

$$Wait_{M_r} \leftarrow Start_{D_r} - D[M_r, F_r] - End_{P_r}, \forall r \in R \quad (4.6.1)$$

$$Wait_{F_r} \leftarrow Start_{P_r} - D[M_{Prec_r}, F_r] - End_{D_{Prec_r}}, \forall r \in R \quad (4.6.2)$$

$$End_{P_r} \leftarrow Start_{P_r} + d_p, \forall r \in R \quad (4.6.3)$$

$$End_{D_r} \leftarrow Start_{D_r} + d_d, \forall r \in R \quad (4.6.4)$$

$$|Start_{P_{r_1}} - Start_{P_{r_2}}| \geq d_p, \forall r_1, r_2 \in R : F_{r_1} = F_{r_2} \quad (4.6.5)$$

$$|Start_{D_{r_1}} - Start_{D_{r_2}}| \geq d_d, \forall r_1, r_2 \in R : M_{r_1} = M_{r_2} \quad (4.6.6)$$

$$End_{D_r} \leq T, \forall r \in R \quad (4.6.7)$$

$$Start_{F_f} \leftarrow \min(Start_{P_r}), \forall r \in R : F_r = f \quad (4.6.8)$$

$$End_{F_f} \leftarrow \max(End_{P_r}), \forall r \in R : F_r = f \quad (4.6.9)$$

$$\text{ALLDIFFERENT}(\text{Prec}) \tag{4.6.10}$$

$$\text{Prec}_r \neq r, \forall r \in R \tag{4.6.11}$$

$$\text{Wait}_{M_r} \geq 0, \forall r \in R \tag{4.6.12}$$

$$\text{Wait}_{F_r} \geq 0, \forall r \in R \tag{4.6.13}$$

Constraints (4.6.12) and (4.6.13) express the fact that at wood mills and forest areas waiting times of trucks are non negative. Constraints (4.6.1) and (4.6.2) express the invariants ( $\leftarrow$ ) related to waiting time of trucks at wood mills and forest areas respectively. Combining (4.6.12) and (4.6.1) ensures that a truck transporting request  $r$  has enough time to do it. Combining (4.6.13) and (4.6.2) ensures that a truck that carries out two successive requests (including the dummy departure) has enough time to do so. Constraints (4.6.3) and (4.6.4) represent the invariants related to the end of time of the pick-up and the delivery activity respectively. Constraints (4.6.5) and (4.6.6) ensure that the log-loaders at forest areas and wood mills have enough time to load and unload any request. Constraints (4.6.8) and (4.6.9) express the invariants computing the starting time of the first pick-up activity and the end of time of the last pick-up activity associated with any forest area. Constraint (4.6.7) specifies that we respect the deadline to transport all requests. Constraint (4.6.10) specifies that all predecessors are given a different value (this combined with the fact that a predecessor variable can only have one value, is equivalent to a flow conservation constraint). Finally, constraint (4.6.11) expresses that a predecessor of a request cannot be the request itself.

#### 4.6.1.5 The Routing Operator

The routing algorithm proposed in this paper and incorporated in the ILS procedure is based on a local search (LS) operator to find a good set of routes, where each

neighborhood represents a set of solutions obtainable from a solution by applying some specified operations. The search space is the set of all order assignments of requests to trucks. The initial solution of the first call of LS is  $(Prec_r = r - nbV, \forall r \in O)$  that satisfies constraints (4.6.10) and (4.6.11).

We use some classical neighborhoods of the VRP in our LS (see Ibaraki et al. [2008]) and limit their size by using parameters; thus five different neighborhoods structures have been implemented in our LS.

- i-opt neighborhood : The i-opt neighborhood (see Bräysy [2003] and Ibaraki et al. [2008]) is used for the travelling salesman problem (TSP) (a variant of the VRP with only one vehicle). It consists in choosing a path (a sequence of some consecutive requests served by the same truck) of at most  $L_{path}^{i-opt}$  (a parameter) logs, and removing it from its position to an other one belonging to the same truck route. We have preserved the same order during the insertion.
- 2-opt neighborhood : The 2-opt neighborhood is a standard neighborhood of the TSP. It consists in choosing a path and inserting it into its current position with the reversed order. In our case, we select two different requests belonging to the same truck route. The 2-opt performs on the path having these two requests as extremities.
- 2-opt\* neighborhood : 2-opt\* is a classical neighborhood of the VRP, it was introduced by Potvin et al. [1996] and consists in selecting two different truck routes dividing each into two parts, and exchanging the second parts of the two routes.
- path insertion neighborhood : It consists in removing a path of at most  $L^{pins}$



(a parameter) requests belonging to a route  $\sigma_k$ , and inserts it into an other route  $\sigma_k^*$ . We have preserved the same order during the insertion.

- cross-exchange neighborhood : Originally proposed by Taillard et al. [1997], it consists in exchanging two different paths of at most  $L^{cross}$  (a parameter) logs belonging to two different truck routes. We have preserved the same order of requests during the insertion.

Among the neighborhoods considered, the last three deal with two different trucks routes,  $\sigma_k$  and  $\sigma_k^*$ . In order to reduce the computational time, we decide to choose  $\sigma_k$  among the 25% of truck routes that have the largest truck waiting times.

#### 4.6.1.6 A Greedy Scheduling Algorithm

The greedy algorithm performs the scheduling of activities (loading and unloading) taking into account the synchronization constraints. The algorithm is described as follows :

- Step 1 : Schedule all activities as soon as possible without considering the fact that loaders are unary ressources, and put all loading activities in a list. Go to step 2.
- Step 2 : Select the earliest loading activity belonging to the list, and if there is any conflict at its related loader, repair it and propagate the correction in the route of the associated truck. Finally, remove this loading activity from the list.
- Step 3 : Select the unloading activity associated with the loading activity of step 2. If there is any conflict at its related loader, repair it and propagate the

correction in the route of the associated truck. Go to step 4.

- Step 4 : If the list of loading activities is not empty, return to step 2.

#### 4.6.1.7 The ILS algorithm

We adopted the ILS based on our LS (see Johnson [1990]), which is one of the basic metaheuristics. In the ILS, the LS is called iteratively. Except the first call of the LS where the initial solution is constructed, other initial solutions related to each call of the LS are generated by slightly perturbing good solutions found so far. Perturbing the current best solution associated with the current iteration of the ILS consists in a random cross exchange operation, which randomly chooses two paths from different routes and exchanges them. In order to generate a new initial solution, we choose  $r$  randomly from  $\{1, 2, 3, 4\}$ , and apply random cross exchange operations  $r$  times consecutively to the best solution found so far. We use the objective function as the evaluation function, and we move to the best solution related to the current neighborhood if it improves the current best solution associated with the current iteration of the ILS.

During the LS processus, neighborhoods are visited consecutively in the order presented before. Once ILS find a better solution in a neighborhood, we return to the i-opt neighborhood.

### 4.6.2 A Hybrid Approach

The decision variables of the CP model are based on the COMET1.1 CP Solver. For each activity  $A$ , two finite domain variables are created,  $A^s$  and  $A^e$ , which are associated respectively to the beginning and the end of the activity.

#### 4.6.2.1 CP Variables and Domains

|                                |   |   |
|--------------------------------|---|---|
| $P_r, \forall r \in O$         | : | A pickup activity of duration $d_p$ if $r \in R$ , and of duration 0 if $r \in (O - R)$ .   |
| $D_r, \forall r \in I$         | : | A delivery activity of duration $d_d$ if $r \in R$ , and of duration 0 if $r \in (I - R)$ . |
| $O_f, \forall f \in F$         | : | An activity of duration equal to the opening time of forest area $f$ .                      |
| $H$                            | : | A dummy task activity of duration 0 that proceeds after all other activities.               |
| $W_r^m \in W, \forall r \in R$ | : | The waiting time of the truck transporting the request $r$ at its related wood mill.        |
| $W_r^f \in W, \forall r \in R$ | : | The waiting time of the truck transporting the request $r$ at its related forest area.      |

#### 4.6.2.2 CP Formulation of the Log-Truck Scheduling Problem

$$\text{Min } \sum_{r \in R} c_d D[M_{Prec_r}, F_r] + \sum_{r \in R} c_t (W_r^f + W_r^m) + \sum_{f \in F} c_l (O_f^e - O_f^s) \quad (4.6.14)$$

$$P_r \Leftrightarrow L^{F_r}, \forall r \in R \quad (4.6.14)$$

$$D_r \Leftrightarrow L^{M_r}, \forall r \in R \quad (4.6.15)$$

$$O_f^s = \min(r \in R : F_r = f) P_r^s, \forall f \in F \quad (4.6.16)$$

$$O_f^e = \max(r \in R : F_r = f) P_r^e, \forall f \in F \quad (4.6.17)$$

$$Prec_{r_2} = r_1 \Rightarrow D_{r_1}^e + D[M_{r_1}, F_{r_2}] \leq P_{r_2}^s, \forall r_1 \in I, \forall r_2 \in O \quad (4.6.18)$$

$$Prec_{r_2} = r_1 \Rightarrow W_{r_2}^f = P_{r_2}^s - D_{r_1}^e - D[M_{r_1}, F_{r_2}], \forall r_1 \in R \forall r_2 \in R \quad (4.6.19)$$

$$P_r^e + D[M_r, F_r] \leq D_r^s, \forall r \in R \quad (4.6.20)$$

$$W_r^m = D_r^s - P_r^e - D[M_r, F_r], \forall r \in R \quad (4.6.21)$$

$$\text{ALLDIFFERENT}(\text{Prec}) \quad (4.6.22)$$

$$Prec_r \neq r, \forall r \in R \quad (4.6.23)$$

$$D_r \prec H, \forall r \in R \quad (4.6.24)$$

$$H^e \leq T, \quad (4.6.25)$$

All constraints of the CP model have been expressed in the same way as in the CBLIS model except constraints (4.6.14) and (4.6.15) that express that loading and unloading require a log-loader for their execution, and constraint (4.6.24) that represents the fact that all requests are served before the dummy task ( $H$ ).

#### 4.6.2.3 CP Search Strategy

The search strategy consists in scheduling the unloading activities as soon as possible by using the procedure `setTimes`, then generating appropriate values for the truck waiting time variables using the procedure `label`. The search space is explored using a depth-first strategy (DFS), which allows to find feasible solutions quickly compared to other strategies. This is important, since the hybridization imposes the best solution found so far as an upper bound in the CP model. Thus, any feasible solution of the CP part outperforms the best CBLIS solution. We limited the global computational time allocated to CP and imposed a restarting strategy after a fixed amount of time by using the procedure (`restartOnTimeLimit`). All these procedures are embedded in the COMET1.1 CP Solver.

#### 4.6.2.4 The Hybrid Algorithm

We propose a hybrid ILS/CP algorithm to solve the daily SLTSP. The idea behind using CP is to measure the efficiency of the greedy algorithm compared to the COMET CP solver, and to improve the solution quality of the greedy algorithm if possible. Since CP consumes more time than the greedy algorithm, we have limited its use in five possible situations, which have been tested independently.

- Call the CP model when the CBLS operator (LS + greedy scheduling algorithm) is unable to improve the best solution found so far.
- Call the CP model when the current ILS move improves the routing and scheduling part of the best solution found so far.
- Use the CP model when the current ILS move improves only the routing part of the best solution.
- Use the CP model only when the scheduling part of the best solution has been improved by the current ILS move.
- Perform the hybridization when the current ILS move deteriorates the routing and scheduling part of the best solution found so far

An ILS move represents a combination of a LS move and a greedy schedule. We note that for all these cases the best solution found so far has been imposed as an upper bound of the objective function in the CP model.

In practice, combining the initial assignment ( $Prec_r = r - nbV, \forall r \in O$ ) and the neighborhood structures  $\{N_1, N_2, N_3, N_4, N_5\}$  allow to remove the constraints (4.6.10), (4.6.11), (4.6.22) and (4.6.23) from the CBLS and CP models respectively. For more details about the ILS and ILS/CP see figure (4.4).

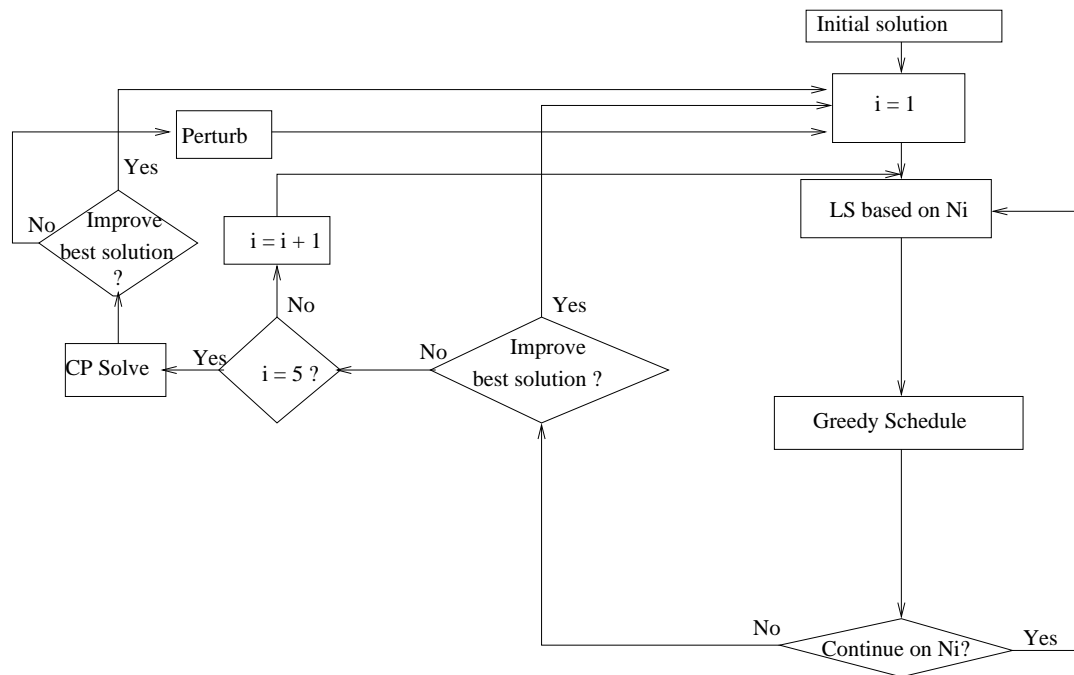


Figure 4.4 – The second phase hybrid algorithm

## 4.7 Experimental Results

We have tested our algorithmic approach on two different industrial case studies provided by the FERIC division<sup>1</sup> of FPInnovations, both of them involving six forest areas and five wood mills. For the first case study, the average cycle time to transport a load is about 4 hours and there are approximatively 400 loads per week, while the second case has an average cycle time of 5.5 hours and approximatively 700 loads per week. In both cases, there are three different products to transport from forest areas to woodmills, and we used the following costs : \$60 per hour for the waiting time of trucks, \$70 per hour for truck travel, and \$100 per hour for the waiting time of log-loaders. For both cases, loading and unloading take about 20 minutes, which motivated our choice to discretize the data in 20 minute-increments, although the loading takes slightly more time. Several tests have been performed by varying the number of trucks, the time allocated to the CP procedure and, finally, the global time allocated to solve the daily Synchronized LTSP. We ran each scenario thirty times and according to our preliminary tests, the first approach, which integrates CBLS and CP when the CBLS operator (LS + greedy scheduling algorithm) is unable to improve the best solution found so far, outperforms all other hybrid approaches. We thus chose to compare the pure ILS algorithm (without hybridization) with this first hybrid approach. We have run the hybrid algorithm allowing the CP procedure to run for 20, 40 or 60 seconds. A comparison of the results obtained indicated that the best value to use was 40 seconds for all scenarios. In the following, we only consider

---

<sup>1</sup>The Forest Engineering Research Institute of Canada is a private, not-for-profit research and development organization whose goal is to improve Canadian forestry operations related to the harvesting and transportation of wood, and the growing of trees, within a framework of sustainable development.

the results obtained for this value.

We report in Tables 4.1 and 4.2 the average total cost, the average routing cost, the average waiting time cost and the worst waiting time cost obtained in 30 runs. Even if CP is time consuming, since it takes 40 seconds at each non-profitable global iteration of the hybrid algorithm, we note that in the second case study ILS/CP outperforms ILS by about 0.5% in terms of the average total cost in all instances except one, by about 0.5% to 2.5% in terms of the average waiting time cost in two thirds of instances, and finally, by about 0.1% to 1.5% in terms of the average routing cost. This reduction in terms of the average waiting is due to the contribution of CP, which was especially adapted to deal with the synchronization constraints, while the reduction reported in terms of the average routing cost is explained by the use of CP as an intensification tool for the local search, which leads the search into regions of the solution space that have not been visited by the ILS. Looking at the first case study (see Table 4.1), we observe that neither of the algorithms (ILS/CP, ILS) is dominant in terms of the average total cost, the average routing cost and the average waiting times cost. Both have a similar behavior. However, for both case studies (see Tables 4.1 and 4.2), the results show that ILS obtains more worst cases in terms of waiting time cost than ILS/CP (more than two thirds of worst cases are associated with the ILS). Furthermore, the introduction of CP (ILS/CP) allows us to obtain more stable results, in the sense of having less variance in the total cost and the waiting time cost (see Table 4.4). Finally, we deduce from the results that both algorithms have a similar behavior in terms of the best solution obtained during the 30 runs (see Table 4.3).

We believe that all these advantages (improvement of average total cost, improvement of the variance and reduction of the worst scenarios) are the result of the combination of ILS and CP. Even if for the first case study, both algorithms have a



similar behavior, the fact that ILS/CP obtains less worst waiting times encourages us to use it, since this part of the solution is directly perceived by the drivers. In all cases where we reported a reduction in the average total cost, we remark that it was not significant (less than 1%). This is explained by the fact that the waiting time cost represents less than 7% of the total cost. Therefore, the CP contribution has an impact on only a small fraction of the overall objective function.

Table 4.1 – **Results of case 1**

| Cost            | nbV | Global time 10 min |        | Global time 15 min |        | Global time 20 min |        |
|-----------------|-----|--------------------|--------|--------------------|--------|--------------------|--------|
|                 |     | ILS                | ILS/CP | ILS                | ILS/CP | ILS                | ILS/CP |
| Average total   | 14  | 61370              | 61223  | 60910              | 61333  | 60940              | 61096  |
|                 | 15  | 61107              | 61130  | 60770              | 60906  | 60780              | 60843  |
|                 | 16  | 60604              | 60253  | 60484              | 60320  | 60436              | 60253  |
| Average routing | 14  | 57306              | 57037  | 56840              | 57223  | 56897              | 56946  |
|                 | 15  | 57241              | 57324  | 56846              | 57070  | 56954              | 56970  |
|                 | 16  | 57054              | 56883  | 56940              | 57010  | 56900              | 56950  |
| Average waiting | 14  | 4064               | 4186   | 4070               | 4110   | 4043               | 4150   |
|                 | 15  | 3866               | 3806   | 3924               | 3836   | 3826               | 3873   |
|                 | 16  | 3550               | 3370   | 3544               | 3310   | 3536               | 3303   |
| Worst waiting   | 14  | 5274               | 5293   | 5120               | 5346   | 5214               | 5080   |
|                 | 15  | 4647               | 4726   | 4534               | 4486   | 4634               | 4393   |
|                 | 16  | 4007               | 3813   | 4007               | 3926   | 4240               | 3686   |

Table 4.2 – Results of case 2

| Cost            | nbV | Global time 10 min |        | Global time 15 min |        | Global time 20 min |        |
|-----------------|-----|--------------------|--------|--------------------|--------|--------------------|--------|
|                 |     | ILS                | ILS/CP | ILS                | ILS/CP | ILS                | ILS/CP |
| Average total   | 30  | 161517             | 161473 | 160050             | 159626 | 158664             | 158316 |
|                 | 31  | 162007             | 159523 | 160760             | 160023 | 159080             | 159203 |
|                 | 32  | 160840             | 160400 | 160504             | 160153 | 160163             | 159726 |
| Average routing | 30  | 153920             | 153693 | 152530             | 152280 | 151428             | 151143 |
|                 | 31  | 154697             | 152247 | 153573             | 152883 | 152090             | 152067 |
|                 | 32  | 153787             | 153284 | 153524             | 153213 | 153243             | 152846 |
| Average waiting | 30  | 7597               | 7780   | 7520               | 7346   | 7236               | 7173   |
|                 | 31  | 7310               | 7276   | 7187               | 7140   | 6990               | 7136   |
|                 | 32  | 7053               | 7116   | 6980               | 6940   | 6920               | 6880   |
| Worst waiting   | 30  | 8394               | 8673   | 8993               | 8533   | 8367               | 8806   |
|                 | 31  | 9554               | 8580   | 9007               | 8480   | 8927               | 8200   |
|                 | 32  | 8400               | 8160   | 8100               | 7866   | 8014               | 7680   |

Table 4.3 – **The best total cost results**

| Case  | nbV | Global time 10 min |        | Global time 15 min |        | Global time 20 min |        |
|-------|-----|--------------------|--------|--------------------|--------|--------------------|--------|
|       |     | ILS                | ILS/CP | ILS                | ILS/CP | ILS                | ILS/CP |
| Case1 | 14  | 59477              | 59734  | 58837              | 59363  | 58540              | 59160  |
|       | 15  | 59747              | 59576  | 58710              | 59260  | 58727              | 59370  |
|       | 16  | 59157              | 58710  | 57383              | 57853  | 58430              | 58043  |
| Case2 | 30  | 154494             | 152730 | 149474             | 151076 | 150223             | 149803 |
|       | 31  | 153587             | 152190 | 153294             | 149033 | 147990             | 149373 |
|       | 32  | 153644             | 155130 | 153050             | 153656 | 152147             | 149056 |

Table 4.4 – **The variance results**

| Time(min) | Average total cost variance (%) |       |        |       | Average waiting time cost variance (%) |       |        |       |
|-----------|---------------------------------|-------|--------|-------|--|-------|--------|-------|
|           | ILS                             |       | ILS/CP |       | ILS                                    |       | ILS/CP |       |
|           | case1                           | case2 | case1  | case2 | case1                                  | case2 | case1  | case2 |
| 10        | 10.2                            | 7.9   | 9.5    | 6.7   | 1.5                                    | 3.7   | 1.3    | 2.6   |
| 15        | 10.5                            | 8.1   | 9.9    | 6.9   | 1.3                                    | 3.4   | 1.3    | 2.9   |
| 20        | 9.7                             | 7.4   | 9.3    | 7.5   | 1.1                                    | 2.6   | 1.1    | 2.3   |

## 4.8 Conclusion

We have presented the weekly log-truck scheduling problem in which one seeks to minimize the sum of transportation and unproductive time costs. To address this problem, we decomposed it into two phases. The first phase determines the destination of full truck loads (from forest areas to wood mills) by minimizing transportation and opened forest area costs. This phase yields seven daily problems covering the whole week, which are then solved sequentially. The objective of the second phase is to schedule the daily transportation of logs by minimizing the cost of unproductive time.

Future research directions involve solving both phases using integer programming software and dealing with additional constraints such as supply constraints, the home bases of trucks and the scheduling of daily lunch breaks.

## CHAPITRE 5 : AN IMPLICIT INTEGER LINEAR PROGRAM TO SOLVE THE WEEKLY LOG-TRUCK SCHEDULING PROBLEM

Dans le chapitre précédent, nous avons présenté une méthode en deux phases pour traiter le SLTSP hebdomadaire. Nous avons remarqué que le nombre de contraintes du modèle est proportionnel au nombre de chargements à effectuer durant la semaine, ce qui nous a motivé pour investiguer d'autres pistes de modélisation où le nombre de contraintes est indépendant des chargements à faire. Nous nous pencherons aussi dans le présent chapitre, sur la question des contraintes de métier (bases de camions et pauses de repas) pour s'approcher davantage de la réalité du terrain.

Cet article propose une nouvelle méthode en deux phases pour traiter le SLTSP hebdomadaire. Outre la politique de la livraison en flux tirés juste à temps permettant l'amélioration de la gestion de stocks des usines déjà prise en compte dans l'article précédent, nous avons pu considérer que les camions sont affectés à des bases régionales à partir desquelles ils commencent leurs journées, et nous avons permis aux camions d'avoir des pauses dîner entre 11h et 15h dans n'importe quelle usine. La première phase est identique à celle de l'article précédent. Elle consiste à minimiser le nombre de sites forestiers durant les jours de la semaine et les coûts de transport plein. Contrairement à ce qui a été fait précédemment, nous avons résolu cette première phase exactement avec Cplex 11. La deuxième phase présente sept problèmes journaliers du lundi au dimanche, modélisés grâce à la programmation linéaire en nombres entiers. Cette modélisation a l'avantage d'avoir une structure proche de celle d'un problème de flot à coût minimum. Elle a permis d'avoir une garantie (Gap) sur la qualité de la solution entière obtenue et, dans certains cas, elle

a prouvé l'optimalité de la solution de nombre de problèmes journaliers.

Cet article a été présenté à la rencontre internationale *CORS-INFORMS2009* tenue à Toronto en juin 2009 et sera dans peu de temps soumis à la revue internationale *European Journal of Operations Research*.

# An Implicit Integer Linear Program to Solve the Weekly Log-Truck Scheduling Problem

El hachemi Nizar,<sup>†‡</sup> El hallaoui Issmail,<sup>†</sup> Gendreau Michel,<sup>†‡</sup>  
and Louis-Martin Rousseau<sup>†‡</sup>

<sup>†</sup> *Ecole Polytechnique de Montréal,  
Département de Mathématiques et Génie Industriel  
C.P. 6079, succursale centre-ville  
Montréal, Québec, Canada H3C 3A7*

<sup>‡</sup> *Interuniversity Research Centre on Enterprise Networks,  
Logistics and Transportation (CIRRELT)  
C.P. 6128, succursale centre-ville  
Montréal, Québec, Canada H3C 3J7*

{nizar.el-hachemi, issmail.el-hallaoui,  
michel.gendreau,louis-martin.rousseau}@polymtl.ca

### Abstract

We present in this paper the weekly log-truck scheduling problem (LTSP) integrating routing and scheduling of trucks when all transports are full truckloads. This problem includes aspects such as pick-up and delivery, multiple products, inventory stock, and lunch breaks for trucks. The objective is to minimize the overall cost including waiting times, loaded travel and empty driven distance. Our solution approach is based on two phases. The first one presents an integer linear program that determines the destinations of full truckloads. The second one is an implicit integer linear program based on an arc formulation model, that ensures routing and scheduling of trucks at a minimum cost. Experiments have been implemented using Cplex 11, and almost all instances were solved within thirty minutes with a reasonable gap.

**Keywords :** Forestry, transportation, routing, scheduling and mixed-integer programming.



## 5.1 Introduction

Forest industry manufactures goods from the timber grown in the forests. It provides a large set of products such as paper, wrappings, building materials and furniture. At the same time, many services were developed around this industry (recreation, preservation of wildlife, species and vegetation). The forest industry occupies an important place in the economy of several countries such as Chile, Canada, Sweden, Finland, New Zealand and Austria. In Canada, this industry employs about 1 million of peoples (255.000 directly and 763.000 indirectly), in 2007 it was worth \$78.3 billion and contributed \$23.4 billion to Canada's trade balance. With respect to the Austrian economy, forest based industry is in the second place in terms of exporting goods and services. Forests cover a large part of Sweden and Finland, which allowed these countries to develop a powerful forest industry that contributes for instance, to about 15.4% in Finland's P.I.B and thus occupies the second place after the electronic sector.

Planning problems in forestry cover a wide scope of activities ranging from planting and harvesting to road building and transportation. In Quebec, transportation represents more than 30% of the cost of provisioning for wood transformation mills, i.e., approximately \$15 per cubic meter of roundwood. The average distance between forest areas where wood is collected and mills to which this wood is transported is around 150 km, and more than 50% of the fuel required per cubic meter of collected wood is consumed by the forest trucks traveling, half of the time empty, between forest areas and mills. It thus follows that transport activities between forest areas and mills should be organized as effectively as possible, both for economic and environmental reasons.

The paper is organized as follows. Sections 5.2, 5.3 and 5.4 present respectively the log-truck scheduling problem, the literature review and the solution approach.

The experimental setting is described in Section 5.5, where computational results are also reported. Section 5.6 concludes the paper.

## 5.2 Log-Truck Scheduling Problem

The log-truck scheduling problem (LTSP) is very closely related to some routing problems encountered in other industries, in particular, so called “Vehicle routing problem” (VRP) and “Pick-up and delivery problems with time windows” (PDPTW) (see Ropke et al. [2007], Savelsbergh et Sol [1995], Cordeau et al. [2002], Gendreau et al. [2002] and the book edited by Toth and Vigo where variants of problem and different algorithms are discussed). In our case, we consider a pick-up and delivery problem such as for each request, exactly one truckload of wood has to be transported from its pick-up location (forest area) to its delivery location (wood mill). A truck visits only one forest area and one wood mill on any given trip, i.e., requests are served individually by trucks. After unloading at a wood mill from its previous trip, a truck is usually sent back empty to its next destination (see Figure 5.1).

It must be noted that the problem as described above differs in several ways from the LTSPs addressed by some other authors. These differences stem from specific characteristics of Canadian forestry operations. Among others, in Canada, cut area and logs volumes are generally quite large. Thus, it is a common practice to express the quantities of wood to be transported in cubic meters which can be easily converted in truckloads. Less than truckload transportation mode is seldomly used in practice. Such as cases arise in Canada when forest industries deal with high value product such as hard wood. Such as case, the expert planners merge similar logs that are going to same destination to assemble full truckloads.

In this paper, we consider a set of wood mills with a number of demands given on a daily basis. A demand is defined by a certain assortment type and the daily

amount required at each wood mill. Similarly, we consider a set of forest areas with a number of supplies where each supply is defined by an assortment type and a weekly available amount. Furthermore, in each wood mill and each forest location, there is a single log-loader that ensures the loading at each forest area and a single one that ensures unloading at each wood mill. When a truck arrives at a location, if the loader is busy, then the truck has to wait until the loader becomes available. When a loader finishes serving a truck, if there is no truck in the line, then the loader has to wait until the arrival of the next truck. These waiting times can severely delay trucks and decrease the usability of the loaders and thus increase the cost of transportation; they should be avoided as much as possible. In our case, the LTSP consists in routing and scheduling trucks integrated with scheduling loaders on a weekly basis. The objective is to minimize unproductive activities (waiting times + deadheads) and full truckloads costs by satisfying some specific constraints :

- Satisfy daily demands;
- Satisfy daily stock constraints;
- Deliver correct assortment to the wood mills;
- The amount picked up at the supply points should not exceed the total supply;
- Precedence constraints : each truck has to pick-up before delivery

In our application, we considered that we have a set of regional bases. Each base is associated with a set of trucks, so that each truck must begin and end at its base in each day. In the case studies, we have considered one and three bases.

### 5.3 Literature Review

Since the mid-1990s, several companies in the forestry sector have initiated major projects aimed at improving the transportation portion of their activities, in particular, the control and quality of truck scheduling. Weintraub et al. [1996] describe a

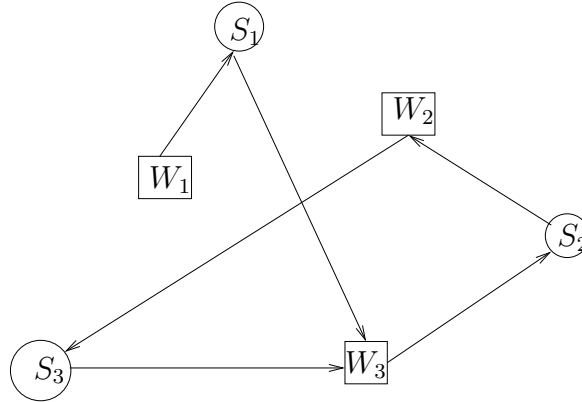


Figure 5.1 – Example of a route for a logging truck

heuristic-based model (ASICAM), developed for Chilean forest companies and used also by some South African forest companies. ASICAM produces a daily plan for each truck by assigning loads and trips, while satisfying supply and demand constraints. The application of ASICAM in Chile has led to a 32% reduction in truck fleet size, a 13% reduction in average working hours and operational costs, and 31% increase in productive hours for one company. Another system called EPO developed by Lin-nainmaa et al. [1995] has been used in Finland. EPO covers all the operations of the wood procurement from strategic planning to actual deliveries to the wood mills. The most challenging part of the system is called EPO2, it determines optimal weekly routes for the trucks, knowing the availability of timber and transportation vehicles. One of the main goal of EPO is the minimization of truck driving as whole. Palmgren et al. [2004] propose a solution method for the LTSP, which is a generalisation of the pick-up and delivery problem with time windows. Their approach is based on column generation and pseudo branch and price, where each column represents one feasible route for one truck. An initial set of routes is generated at the beginning by using the heuristic described in Palmgren et al. [2003]. Then, the subproblem, which is a constrained shortest path problem, is solved by applying a k-shortest path

algorithm. Gronalt and Hirsch [2007] applied a modified version of the Unified Tabu Search heuristic to solve a restricted variant of the LTSP in which the number of trips between each forest area and each wood mill is given. The same assumption is used by El Hachemi et al. [2008] who develop a hybrid method combining Integer programming (IP) and Constraint Programming (CP) where the IP model generates optimal routes in term of deadheading, while CP deals with the scheduling part and takes into account the synchronization constraints between trucks and loaders. Flisberg et al. [2009] and Andersson et al. [2008] proposed a two-phase solution approach that transforms the LTSP into a standard vehicle routing problem with time windows. The first phase determines the flow of wood from supply points to demand points, while the second phase combines transport nodes into routes. The dispatching procedure that continuously updates the trucks routes during the day is based on the previous work of Rönnqvist et Ryan [1995] and Rönnqvist et al. [1998] that generates one trip at a time.

With respect to numerical experiments, Murphy [2003] presents a case study with an average 9 trucks and 35 transport tasks per day, while Palmgren et al. [2003] have solve two case studies for Sweden forest industry : one with 6 trucks and 39 transport tasks, and another one with 28 trucks and approximately 85 transport tasks. Gronalt and Hirsch [2007] solved random problems with 30 transport tasks and 10 trucks. El Hachemi et al. [2008] deal with daily problems having up to 18 trucks and 70 transport tasks. Finally, Flisberg et al. [2009] and Andersson et al. [2008] have solved substanstially larger instances ranging from 188 transport tasks to about 2.500 full truckloads with 15 to 110 trucks. In this paper we solve weekly problems having up to  $\approx 700$  transport tasks. For a more detailed description of optimization problems in the forest sector, we refer the reader to Rönnqvist [2003] and D'Amours et al. [2008] .

There are many applications in which vehicles must be synchronized. This occurs

in distribution systems when a vehicle cannot pick-up a load until another vehicle has first delivered it, or in urban mass transit systems when drivers have to change buses at so-called relief points (see Freling et al. [2003] and Haase et al. [2001]). Bredström and Rönnqvist [2008] present a mathematical programming model for the combined vehicle routing and scheduling problem with additional precedence and synchronization constraints. To illustrate the practical significance of the temporal precedence and synchronization constraints, the authors test their approach with the homecare staff scheduling problem (see Everborn et al. [2006] and Everborn et al. [2009]) where, in some customer visits, it requires that two staff members occurs simultaneously or in given order. They discuss the importance of synchronization constraints in the airline industry, for example in a modified version of the fleet assignment problem, there is a need to state that the same flight must departure on the same time on each day to get a robust schedule. They also discuss the need of coordination between harvesters and forwarders in the forestry context. Since forwarding can only be done once harvesting has been performed. With respect to the LTSP, the authors discuss the requirement to synchronize trucks (without crane) and loaders, since these loaders often serve several stands and move between these in order to load trucks.

## 5.4 Solution Approach

In this paper, we focus on developing a solution method to solve a modified version of the weekly LTSP allowing the wood mills to operate in just in time mode. El Hachemi et al. [2009] propose a hybrid local search (LS)/CP method to deal with this problem. However, new challenges as lunch breaks, supply constraints and home bases of trucks motivated us to search a new model adapted to the daily problem, and representing as much as possible the reality, and being faster than the LS/CP

approach.

A two phase approach has been used in this paper. The first one which we call “tactical problem” consists in solving an IP formulation taking into account demand, supply and stock constraints. In this step, restrictions are on time availability of trucks. This phase yields seven daily LTSP (from Monday to Sunday). For each daily problem, there is a fixed set of transportation requests to perform. The implicit models have proven their effectiveness in solving multiple variants of the shift scheduling problem (see Aykin [1996], Bechtolds and Jacobs [1990], and Rekik et al. [2004]), this has motivated our research in this direction. We propose an implicit flow based IP model to address these issues, where each component of a truck trip has been represented in the network as an arc. We used Cplex 11 to solve the two phases.

#### **5.4.1 The Tactical Problem**

In this part, we present our tactical planning model where we take into consideration the fact that different wood products must be shipped to wood mills. These multi-products demand constraints rise from the fact that many wood mills order logs in specific lengths and diameters to produce given final products and the fact that the properties of wood are strongly related to the particular tree that yields it, i.e., for example, softwood in general is preferred over hardwood when wood is used to manufacture lumber, plywood and other wood-based panels. Thus, the logs are sorted into different assortments that depend on species, usage, quality and dimension. Each supply points consists of a given assortment group (up to 3 products in our case) and each demand point present a requirement of a given assortment group. In general, the inventory is known at the beginning of the week since it is the stock associated with the last day of the previous week. In some cases, when the demands at wood mills and supplies at forest areas remain constant during a long period co-

vering many weeks, it is judicious to generate a weekly solution that can be repeated in the whole period. To achieve this goal, we impose that inventory of the first day is equal to the inventory of the last day of the week.

#### 5.4.1.1 Parameters

|            |   |
|------------|---|
| $F$        | : the set of forest areas.  |
| $M$        | : the set of wood mills.  |
| $P$        | : the set of forest products.   |
| $J$        | : the set of days of the week.  |
| $S_p^l$    | : the daily maximum number of full truckloads of product $p$ that can be stocked per wood mill.                         |
| $U$        | : the daily upper bound on the number of full truckloads that can be transported from any forest area.                  |
| $L$        | : the daily lower bound on the number of full truckloads that must be transported from any forest area if it is opened. |
| $D_{mp}^j$ | : the demand of product $p$ at wood mill $m$ , in day $j$ .   |
| $S_{fp}$   | : the supply of product $p$ at forest area $f$ in the week.   |
| $c_{fm}$   | : the travelling cost between forest area $f$ and wood mill $m$ .   |
| $C^*$      | : the daily fixed cost of opening any forest area.  |

#### 5.4.1.2 Variables

|             |   |
|-------------|---|
| $O_f^j$     | : A binary variable equal to 1 if forest area $f$ is opened day $j$ .   |
| $T_{fmp}^j$ | : An integer variable representing the number of full truckloads of product $p$ from supply point $f$ to wood mill $m$ on day $j$ . |
| $K_{mp}^j$  | : An integer variable representing the number of full truckloads of product $p$ stocked at wood mill $m$ on day $j$ .               |

#### 5.4.1.3 The tactical model

The IP formulation of the tactical problem is as follows :

$$\begin{aligned} \text{Min} \quad & \sum_{j \in J} \sum_{f \in F} C^* O_f^j + \sum_{j \in J} \sum_{f \in F} \sum_{m \in M} c_{fm} T_{fmp}^j \\ \text{subject to} \quad & \end{aligned}$$



$$K_{mp}^j \leq S_p^l, \forall m \in M, \forall p \in P, \forall j \in J \quad (5.4.1)$$

$$K_{mp}^{j-1} + \sum_{f \in F} T_{fmp}^j = D_{mp}^j + K_{mp}^j, \forall m \in M, \forall p \in P, \forall j \in J \quad (5.4.2)$$

$$\sum_{j \in J} \sum_{m \in M} T_{fmp}^j \leq S_{fp}, \forall f \in F, \forall p \in P \quad (5.4.3)$$

$$\sum_{m \in M} \sum_{p \in P} T_{fmp}^j \leq U \cdot O_f^j, \forall f \in F, \forall j \in J \quad (5.4.4)$$

$$\sum_{m \in M} \sum_{p \in P} T_{fmp}^j \geq L \cdot O_f^j, \forall f \in F, \forall j \in J \quad (5.4.5)$$

In this part, the objective function is to minimize the sum of the opening cost of forest areas and the full truckloads cost and finally. We decide to set  $C^* \gg c_{fm}$ , so that the optimizer seeks to minimize the number of sites in operation during the week. This will induce global control over daily log-loaders waiting times, since no costs are attributed in this model to inactive loader during the whole day.

Constraint (5.4.1) ensures that the daily stock of any product at any wood mill respects the maximum limit fixed. Constraint (5.4.2) specifies that demand is satisfied over the whole week. Constraint (5.4.3) expresses the fact that we respect the amount of supply available of each product at each forest area. Constraint (5.4.4) represents the fact that each loader has a finite capacity to load logs (a loader cannot exceed a number of loads per day). Constraint (5.4.5) indicates that if a loader is operating during one day, it must work at least for some minimum time.

## 5.4.2 Phase 2 : The Daily Synchronized Log-Truck Scheduling Problem

We present an implicit IP model to deal with the daily synchronized log-truck scheduling problem (SLTSP) that is very close to a network flow formulation. Each component (activity) of a truck trip (deadhead, loading, loaded travel, unloading) is

modeled as an arc in a network, as well as truck waiting times (see Figure 5.2). We decide to represent waiting arcs only at wood mills in order to avoid symmetry, since we have less wood mills than forest areas. For each regional base, we assume that we have a homogeneous fleet of trucks.

### 5.4.2.1 Parameters

|               |  |
|---------------|--|
| $B$           | : the set of all bases.  |
| $c_l$         | : the cost of waiting time of a log-loader per unit of time.   |
| $c_a$         | : the cost associated with arc $a$ (empty driven arcs and trucks waiting times).   |
| $t_l$         | : the loading time of one shipment.  |
| $t_u$         | : the unloading time of one shipment.  |
| $H$           | : the optimization horizon.  |
| $T_a^s$       | : the starting time associated with arc $a$ .  |
| $T_a^e$       | : the ending time associated with arc $a$ .  |
| $trip_{fm}$   | : the number of full truckloads to perform between forest area $f$ and wood mill $m$ .   |
| $L_{(f,m)}^b$ | : the set of loaded trip arcs linking forest area $f$ and wood mill $m$ and associated with base $b$ .   |
| $L^b$         | : $\cup_{f \in F} \cup_{m \in M} L_{(f,m)}^b$ the set of loaded trip arcs associated with base $b$ .   |
| $A^+(s_b)$    | : the set of exiting arcs from the source node $s_b$ associated with base $b$ .  |
| $A^-(t_b)$    | : the set of entering arcs into the sink node $t_b$ associated with base $b$ .   |
| $A^+(i)$      | : the set of exiting arcs from the node $i$ .  |
| $A^-(i)$      | : the set of entering arcs into the node $i$ .   |
| $A_{f,b}^T$   | : the set of loading arcs ( $a_b$ ) loading at forest area $f$ and associated with base $b$ , such that their starting times $T_{a_b}^s \in [T, T + t_l[$ . $T \leq H$ .   |
| $A_{m,b}^T$   | : the set of unloading arcs ( $a_b$ ) unloading at wood mill $m$ and associated with base $b$ , such that their starting times $T_{a_b}^s \in [T, T + t_u[$ . $T \leq H$ . |
| $C_f^b$       | : the set of loading arcs associated with forest area $f$ and base $b$ .   |
| $C^b$         | : $\cup_{f \in F} C_f^b$ the set of loading arcs associated with base $b$ .  |
| $D^b$         | : the set of unloading arcs associated with base $b$ .   |
| $E^b$         | : the set of empty driven arcs associated with base $b$ .  |
| $W^b$         | : the set of truck waiting arcs associated with base $b$ .   |
| $N^b$         | : the set of nodes associated with base $b$ .  |
| $nbV^b$       | : the number of trucks of base $b$ .   |

### 5.4.2.2 Variables

- $x_{a_b}$  : A binary variable if arc  $a_b \in \cup_{b \in B} C^b \cup D^b \cup E^b \cup L^b$  and an integer variable bounded by  $nbV^b$  if  $a_b \in W^b$ .  
 $t_f^{min}$  : the starting time of log-loader associated with forest area  $f$ .  
 $t_f^{max}$  : the ending time of log-loader associated with forest area  $f$ .

### 5.4.2.3 Flow based model

$$\begin{aligned} \text{Min} \quad & \sum_{b \in B} \sum_{a_b \in E^b \cup W^b} c_{a_b} x_{a_b} + \sum_{f \in F} c_l (t_f^{max} - t_f^{min}) \\ \sum_{a_b \in A^+(s_b)} x_{a_b} \quad &= nbV^b, \forall b \in B \end{aligned} \quad (5.4.6)$$

$$\sum_{a_b \in A^+(s_b)} x_{a_b} = \sum_{a_b \in A^-(t_b)} x_{a_b}, \forall b \in B \quad (5.4.7)$$

$$\sum_{a_b \in A^+(i_b)} x_{a_b} = \sum_{a_b \in A^-(i_b)} x_{a_b}, \forall b \in B, \forall i_b \in N_b - \{s_b, t_b\} \quad (5.4.8)$$

$$\sum_{b \in B} \sum_{a_b \in A_{f,b}^T} x_{a_b} \leq 1, \forall f \in F, \forall T \leq H \quad (5.4.9)$$

$$\sum_{b \in B} \sum_{a_b \in A_{m,b}^T} x_{a_b} \leq 1, \forall m \in M, \forall T \leq H \quad (5.4.10)$$

$$(H - T_{a_b}^s) x_{a_b} \leq H - t_f^{min}, \forall b \in B, \forall f \in F, \forall a_b \in C_f^b \quad (5.4.11)$$

$$T_{a_b}^e x_{a_b} \leq t_f^{max}, \forall b \in B, \forall f \in F, \forall a_b \in C_f^b \quad (5.4.12)$$

$$\sum_{b \in B} \sum_{a_b \in L_{(f,m)}^b} x_{a_b} = trip_{fm}, \forall f \in F, \forall m \in M \quad (5.4.13)$$

$$x_{a_b} \in \{0, 1\}, \forall b \in B, \forall a_b \in C^b \cup D^b \cup E^b \cup L^b \quad (5.4.14)$$

$$x_{a_b} \in \{0, 1, 2, \dots, nbV^b\}, \forall b \in B, \forall a_b \in W^b \quad (5.4.15)$$

The objective function is to minimize the daily deadheads and waiting time cost of trucks and the forest log-loaders. Constraint (5.4.6) allows to respect the available fleet of vehicles related to each regional base. Constraints (5.4.7) and (5.4.8) are

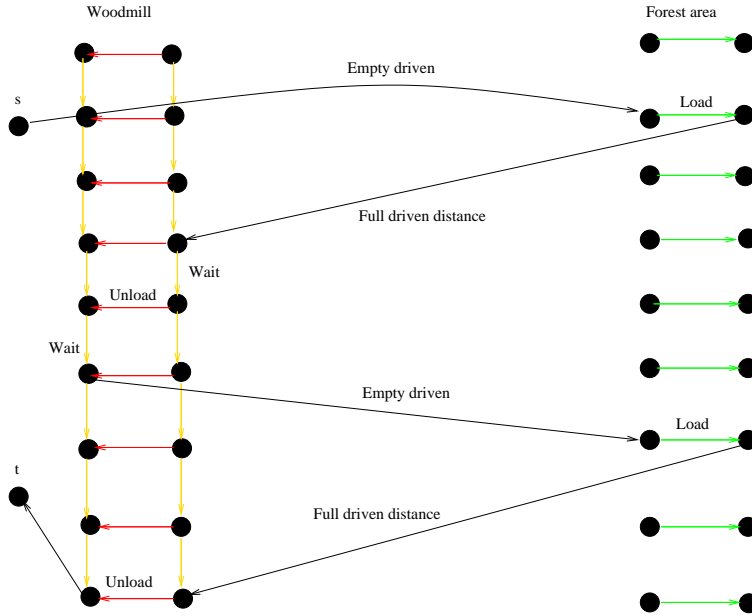


Figure 5.2 – Network Arcs Formulation Model

flow conservation constraints related to each base. Constraints (5.4.9) and (5.4.10) ensure that each loader is serving one truck at once. Constraints (5.4.11) and (5.4.12) compute the starting time and ending time of each forest log-loader. Constraints (5.4.13) indicate the daily number of requests to perform. Finally, constraints (5.4.14) and (5.4.15) express the fact that except waiting time arcs all the others have a capacity of 1.

The forest companies express the need to ensure for each truck a break of one hour between 11h and 15h at any wood mill, where drivers can take their lunches and trucks can be refueled. To satisfy this new constraint, we divide the network in two parts at wood mills nodes, one before the break and one after. The arcs linking both of parts represent break activities of trucks and have a duration of one hour as specified by the forest companies (See Figure 5.3).

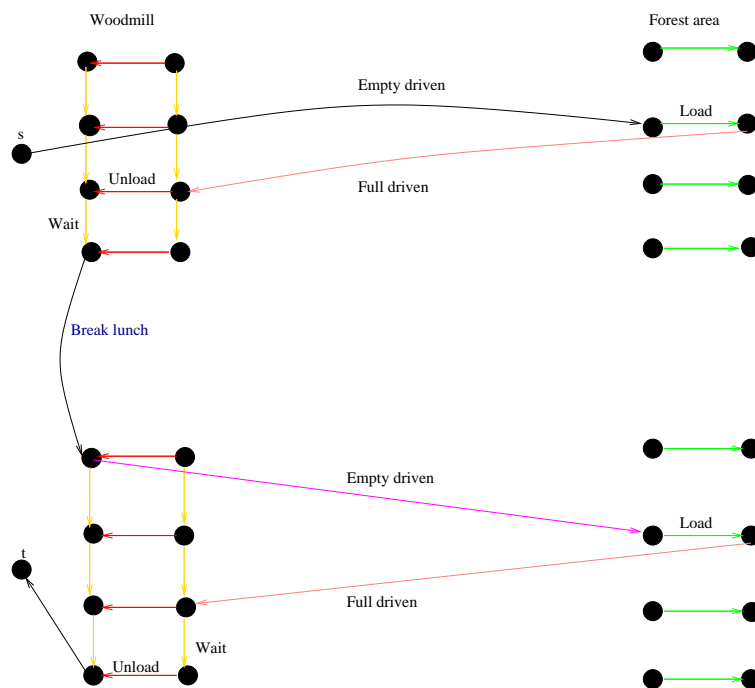


Figure 5.3 – Enriched Network Model

#### 5.4.2.4 Discretization

The discretization step is one of the basic and important parameter of the model, since the precision of solution is strongly related to it. But one of the major interests of our study is that we can exploit the fact that loading and unloading times are approximatively equal (around 20 minutes, even if, in practice, loading takes a little bit more time than unloading). This property allows us to choose the discretization step equal to the loading time, thus it leads to simplify or even remove constraints (5.4.9) and (5.4.10) from the model, especially, in the case where we have only one regional base, making sure that each log-loader can not serve two different trucks or more at the same time is automatically enforced by the flow conservation constraints, and the unit capacity on the loading arc.

#### 5.4.2.5 Branching strategy

We developed a branching strategy that we called a “Start as Late End as Early” (SLEE) based on reducing forest loaders waiting time cost in order to minimize the waiting time cost component of the objective function. We focused on forest loaders waiting time since one hour log-loader waiting time costs approximately the double compared to a truck. To address this issue, we put forward selecting two loading arcs (variables) belonging to the same forest area in two directions (see Figure 5.4), so that we favor branching alternatively on the earliest and the later fractional loading arcs (variables). This strategy leads to reduce the working time for each forest log-loader, and since the daily requests in the second phase are known in advance (a result of phase 1), this will induce the minimization of forest log-loaders waiting times.

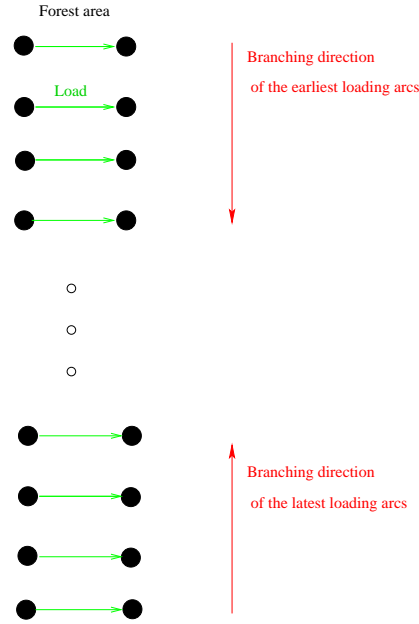


Figure 5.4 – Branching Strategy

## 5.5 Experimental Results

We received two different case studies provided by Fpinnovations (FERIC<sup>1</sup> division). Both of them involve six forest areas and five wood mills. The first case study counts approximately 400 shipments (logs) per week, and has an average cycle time of about 4 hours to transport a shipment, while the second case counts approximately 700 shipments per week with an average cycle time of about 5,5 hours to transport a shipment. Both cases count three different products to transport. We receive from Fpinnovation an approximation of costs adapted to the case studies. In these test

---

<sup>1</sup>The Forest Engineering Research Institute of Canada is a private, not-for-profit research and development organization whose goal is to improve Canadian forestry operations related to the harvesting and transportation of wood, and the growing of trees, within a framework of sustainable development.



cases each truck costs \$60 per hour if it is waiting and \$70 if it is travelling. However, each loader costs more than a truck : \$100 for one hour when it is waiting. Each case has three different fleet sizes (from 14 to 16 trucks for the first case and from 30 to 32 for the second one). For each of these scenarios, we performed three tests, the first one with only one regional base, the second one with three regional bases and the last one with a home base associated with each truck. For each instance, we used two step-sizes (10 minutes and 20 minutes). Finally, all daily SLTSPs have been run exactly for 3 minutes initially and then 10 minutes. In the first phase, we fixed the computational time to 5 minutes, and we report that the first case study has been solved optimally within less than 1 minute and half, while for the second case we obtain a solution presenting 4.5% of gap.

We compare the default Cplex 11 branching strategy with the SLEE strategy related to the proposed previous model (arc formulation model). Looking at all tables, and by considering that one strategy is better than the other if its total cost is at least 0.1% less than the other one, we notice that globally in 55% of instances the SLEE strategy outperforms Cplex 11 default branching strategy, whereas the opposite is true only for 14% of instances. Looking at the results step-size by step-size, and based on Tables 5.1 and 5.3 and Tables 5.2 and 5.4, we remark that the SLEE strategy is clearly better than the Cplex 11 default branching strategy with 10 minutes step-size, while this superiority is not clearly perceived with 20 minutes step-size since both methods are equivalent for 44% of related instances. Although, for the same number of cases (44% of instances) we obtained that the SLEE strategy outperforms the default one. To explain this, we believe that the fact of choosing 20 minutes as a step-size simplifies the problem in the sense that we have fewer arcs (variables) and especially, fewer constraints. Thus, the simplicity in the model allows the two methods to be quasi equivalent (particularly for a large computational time 10 minutes) with a slight advantage for the SLEE strategy. The last observation leads us

to consider the comparison between the two methods with respect to the computational time. Looking at tables 5.1, 5.2, 5.3 and 5.4, we notice that for 3 minutes of computational time for each daily problem, the SLEE strategy is absolutely better than the default one (in 70% of cases). However, once again the difference between the two methods is balanced when we have 10 minutes of computation time. Thus, we can state that the SLEE strategy leads quickly to good quality solution. But, with enough computational time, this strategy can become aggressive in the sense of reducing working time of each forest loader, and this will lead to a deterioration of the routing part. We believe it is that what happens in the instances where the default strategy outperforms the SLEE one. To conclude the comparison between the two branching methods, we remark that the SLEE strategy has led in almost all tests to the solution with the lowest cost in terms of waiting time, this is clearly the consequence of promoting branching reducing the forest-loaders waiting times.

Theoretically, the optimal solution associated with 10 minutes step-size must be better than the optimal solution with 20 minutes, such as the case of the results of the second case study with one regional base where the best solution found with the first step-size (10 minutes) outperforms the second one. However, in other scenarios, the opposite happened, this is explained by the fact that the solutions found with 10 minutes step-size are a little far from the optimum compared to those of the 20 minutes step-size, (the gap clearly confirms this deduction, see tables 5.1, 5.2, 5.3 and 5.4). This difference in the solution quality between the two step-sizes is not surprising, since the 10 minutes step-size model counts more variables (arcs) and constraints. Hence, for these reasons the linear relaxation of the 10 minutes discretization takes more time to solve and consequently, we explore less branching nodes within the time limit with the first step of discretization. Looking at the behavior of the gap, we find that it increases when the number of bases increases, which is normal since the problem size (constraints and variables) also increases. We observed

that the average gap decreases in the big instances (cas study 2) compared to small instances (cas study 1) (see tables 5.1, 5.2, 5.3 and 5.4). Bramel and Simchi-Levi [1997] have discussed for the set covering formulations for the vehicle routing problem with time windows the impact of the size of instances on the gap. They demonstrate that the relative gap between fractional and integer solutions of the problem becomes arbitrarily small as the number of customers increases.

As said at the beginning, our tests are divided in three categories, the first one is associated with a single regional base, the second one associated with three regional bases and the last one associated with a base for each truck. Concerning this last family, we report that we were unable to find any feasible solution for all instances, except the scenarios with 10 minutes of computational time and 20 minutes step-size of the first case study. We believe that with the constraints imposing that each truck must have its related base, the arc flow formulation becomes too large and the model is no longer implicit. We believe that it would be better to consider a column generation approach in this case to take advantage from the problem structure and especially, the block diagonal matrix structure.

The main weakness of our approach consists in the fact that the more we refine the discretization of data, the more the model size increases (since both number of arcs and constraints increase). Finding a technique to generate less arcs and constraints when we refine the discretization of data is thus the subject of future research.

## 5.6 Conclusion

In this paper, we described a two phase method to solve the weekly Log-Truck Scheduling Problem, that includes the just-in-time delivery policy, which is of great interest to the industry.

Table 5.1 – Results with one base and 20 minutes step discretization

| Case  | nbV | Time(min) | Default branching strategy |            |          | SLEE strategy |            |          |
|-------|-----|-----------|----------------------------|------------|----------|---------------|------------|----------|
|       |     |           | Gap %                      | Waiting \$ | Total \$ | Gap %         | Waiting \$ | Total \$ |
| Case1 | 14  | 3         | 5.9                        | 3833       | 66316    | 4.7           | 2766       | 65300    |
|       |     | 10        | 4.5                        | 2700       | 65323    | 4.4           | 2533       | 65113    |
|       | 15  | 3         | 4.4                        | 2733       | 66066    | 4.1           | 2700       | 65863    |
|       |     | 10        | 3.2                        | 2100       | 65310    | 3.4           | 1966       | 65270    |
|       | 16  | 3         | 3.2                        | 2166       | 66190    | 3.1           | 1866       | 66010    |
|       |     | 10        | 2.2                        | 1667       | 65623    | 2.7           | 1633       | 65776    |
| Case2 | 30  | 3         | 1.3                        | 4060       | 162320   | 1.3           | 4033       | 162330   |
|       |     | 10        | 1.2                        | 4060       | 162303   | 1.3           | 4000       | 162326   |
|       | 31  | 3         | 1.1                        | 3387       | 163130   | 1.1           | 3306       | 163106   |
|       |     | 10        | 0.8                        | 3220       | 162934   | 1             | 3206       | 163006   |
|       | 32  | 3         | 0.8                        | 3386       | 164063   | 0.9           | 2986       | 164036   |
|       |     | 10        | 0.7                        | 3286       | 163960   | 0.8           | 2920       | 163970   |

Table 5.2 – Results with one base and 10 minutes step discretization

| Case  | nbV | Time(min) | Default branching strategy |            |          | SLEE strategy |            |          |
|-------|-----|-----------|----------------------------|------------|----------|---------------|------------|----------|
|       |     |           | Gap %                      | Waiting \$ | Total \$ | Gap %         | Waiting \$ | Total \$ |
| Case1 | 14  | 3         | 9.8                        | 7210       | 69306    | 9.1           | 6710       | 68933    |
|       |     | 10        | 8.1                        | 5633       | 67990    | 7.8           | 5666       | 67920    |
|       | 15  | 3         | 7.8                        | 5266       | 68616    | 7.2           | 5253       | 68270    |
|       |     | 10        | 5.6                        | 3033       | 66653    | 6.1           | 4400       | 67353    |
|       | 16  | 3         | 7.2                        | 5150       | 68973    | 5.1           | 3533       | 67333    |
|       |     | 10        | 5.7                        | 3783       | 67720    | 4.5           | 3133       | 66880    |
| Case2 | 30  | 3         | 2                          | 4043       | 162140   | 2             | 4016       | 162206   |
|       |     | 10        | 1.8                        | 3826       | 161923   | 1.9           | 3816       | 162006   |
|       | 31  | 3         | 1.8                        | 3750       | 162833   | 1.6           | 3263       | 162433   |
|       |     | 10        | 1.6                        | 3616       | 162643   | 1.6           | 3246       | 162416   |
|       | 32  | 3         | 1.7                        | 3840       | 163830   | 1.4           | 3276       | 163343   |
|       |     | 10        | 1.4                        | 3473       | 163406   | 1.4           | 3176       | 163243   |

Table 5.3 – Results with three bases and 20 minutes step discretization

| Case  | nbV | Time(min) | Default branching strategy |            |          | SLEE strategy |            |          |
|-------|-----|-----------|----------------------------|------------|----------|---------------|------------|----------|
|       |     |           | Gap %                      | Waiting \$ | Total \$ | Gap %         | Waiting \$ | Total \$ |
| Case1 | 14  | 3         | 9.6                        | 6946       | 69233    | 9             | 6233       | 68710    |
|       |     | 10        | 7.6                        | 5146       | 67573    | 7.6           | 5133       | 67560    |
|       | 15  | 3         | 8.1                        | 6073       | 69016    | 7.8           | 5666       | 68880    |
|       |     | 10        | 6.8                        | 4933       | 67880    | 7.2           | 5366       | 68336    |
|       | 16  | 3         | 7.1                        | 5566       | 69173    | 6.7           | 5000       | 68746    |
|       |     | 10        | 5                          | 3633       | 67356    | 5.1           | 3633       | 67356    |
| Case2 | 30  | 3         | 2.4                        | 4480       | 163013   | 2.2           | 4246       | 162663   |
|       |     | 10        | 2                          | 4133       | 162430   | 1.9           | 3600       | 162146   |
|       | 31  | 3         | 2                          | 4226       | 163486   | 1.7           | 3606       | 162973   |
|       |     | 10        | 1.7                        | 3833       | 162913   | 1.7           | 3753       | 162913   |
|       | 32  | 3         | 1.7                        | 3920       | 164046   | 1.6           | 4000       | 163933   |
|       |     | 10        | 1.5                        | 3500       | 163780   | 1.5           | 3280       | 163750   |

Table 5.4 – Results with three bases and 10 minutes step discretization

| Case  | nbV | Time(min) | Default branching strategy |            |          | SLEE strategy |            |          |
|-------|-----|-----------|----------------------------|------------|----------|---------------|------------|----------|
|       |     |           | Gap %                      | Waiting \$ | Total \$ | Gap %         | Waiting \$ | Total \$ |
| Case1 | 14  | 3         | 12.6                       | 9643       | 71810    | 13.7          | 10340      | 72560    |
|       |     | 10        | 10.6                       | 7733       | 69960    | 11.2          | 8286       | 70333    |
|       | 15  | 3         | 10.9                       | 8226       | 71173    | 10.9          | 8093       | 71016    |
|       |     | 10        | 9.6                        | 7363       | 70296    | 9.6           | 7153       | 70106    |
|       | 16  | 3         | 9.3                        | 7066       | 70873    | 9.3           | 7133       | 70743    |
|       |     | 10        | 7.8                        | 5876       | 69560    | 7.7           | 5710       | 69356    |
| Case2 | 30  | 3         | 4.5                        | 8136       | 166333   | 3.6           | 6270       | 164603   |
|       |     | 10        | 2.7                        | 5063       | 163033   | 2.6           | 4953       | 162980   |
|       | 31  | 3         | 3.8                        | 6883       | 166086   | 3.9           | 7070       | 166136   |
|       |     | 10        | 2.4                        | 4653       | 163656   | 2.1           | 4213       | 163100   |
|       | 32  | 3         | 3.2                        | 5706       | 166143   | 2.9           | 5650       | 165706   |
|       |     | 10        | 1.8                        | 3676       | 163723   | 2.1           | 4280       | 164326   |

Table 5.5 – **Results with a base for each truck and 20 minutes step discretization**

| Case  | nbV | Time(min) | Default branching strategy |            |          | SLEE strategy |            |          |
|-------|-----|-----------|----------------------------|------------|----------|---------------|------------|----------|
|       |     |           | Gap %                      | Waiting \$ | Total \$ | Gap %         | Waiting \$ | Total \$ |
| Case1 | 14  | 3         | –                          | –          | –        | –             | –          | –        |
|       |     | 10        | 11                         | 8266       | 70583    | 10.8          | 7686       | 70363    |
|       | 15  | 3         | –                          | –          | –        | –             | –          | –        |
|       |     | 10        | 9.3                        | 7167       | 70126    | 9.3           | 7100       | 69956    |
|       | 16  | 3         | –                          | –          | –        | –             | –          | –        |
|       |     | 10        | 7.4                        | 5833       | 69416    | 7.5           | 6000       | 69536    |

The first phase determines the optimal full truckloads destinations (from forests area to wood mills) by minimizing the loaded transportation and opening forest areas costs and solved by the Cplex 11 default branching strategy. This phase yields to seven daily SLTSPs. To deal with this, we developed a model called “arcs formulation model” and a branching strategy called “SLEE”. In order to increase accuracy of the data, we reduce the discretization step. Consequently, we increase the complexity of the linear relaxation of the model. Computational experiments show that the SLEE strategy provides higher quality of solutions rapidly compared to the Cplex 11 default strategy.

Other research directions involve solving the problem in one step, using the lagrangian relaxation approach by relaxing all constraints that link two successive days. We are planning to produce robust solutions by including stochasticity in displacements times, loading and unloading times. We orient our research to deal with additional constraints as the drivers rests or changeovers and a home base of each



truck, we believe that column generation approach may be beneficial in this context. Finally, we will try to solve the daily problem continuously during the day in a real time approach.

## CHAPITRE 6 : CONCLUSION

Dans cette thèse, nous nous sommes intéressé au problème de construction d’horaire de camions et chargeuses en contexte forestier SLTSP appliqué à l’industrie forestière canadienne. Il s’agit d’un problème de transport intégrant la tournée et l’horaire de camions avec celle des chargeuses en tenant compte des contraintes de synchronisation (entre camions et chargeuses) pour réduire les coûts d’attentes des camions et des chargeuses en forêts. C’est l’apport prédominant de notre recherche.

Nous avons établi trois modèles différents associés au SLTSP, un modèle PPC, un autre RLBC et finalement un modèle en nombres entiers. Les deux premières contributions de cette thèse se sont focalisées sur la PPC, puisque celle-ci s’est avérée efficace pour traiter les problèmes d’ordonnancement durant les deux dernières décennies.

Au chapitre 3, nous avons présenté une décomposition naturelle du SLTSP journalier séparant la partie tournée de celle d’horaire. Ainsi, nous avons développé un modèle en nombres entiers, basé sur un flot à coût minimum dans un réseau, permettant de gérer la partie tournée des camions en minimisant les coûts des voyages à vide, sachant que, dans la réalité, les camions commencent et finissent leur journée dans une base régionale. Nous avons relaxé cette contrainte de façon à ce que les véhicules puissent commencer et finir leur travail journalier à n’importe quelle usine. Le modèle PPC décrit dans ce même chapitre assure l’horaire des camions et chargeuses en tenant compte des contraintes de synchronisation. Précisons que, au niveau du problème journalier, nous avons veillé à relaxer les contraintes de changements de chauffeurs effectués après 12 heures de conduite et celles relatives aux pauses repas en usines. La communication (hybridation) entre les deux modèles est établie via des contraintes de cardinalité dites globales (GCC). À notre connaissance, c’est la

première fois qu’une telle hybridation est réalisée, ce qui constitue notre contribution majeure dans cette première partie. Il s’est avéré parfois que la structure de tournée qu’on impose peut induire beaucoup d’attente. Pour éviter ces cas, nous avons introduit de la perturbation via un coefficient de pénalité dans le modèle en nombres entiers, tout en préservant sa structure de flot à coût minimum et en gardant les mêmes mécanismes de communication entre les deux modèles.

Dans le chapitre 4, nous avons résolu la version hebdomadaire du SLTSP en deux phases. La première permet d’établir les voyages pleins de la semaine en ayant comme objectif principal la minimisation du nombre de sites ouverts et le coût des voyages pleins. À la sortie de cette phase, sept SLTSPs journaliers sont générés du lundi au dimanche. Dans la phase deux, nous avons modélisé le SLTSP journalier de deux manières : la première méthode reprend exactement le modèle PPC de la première contribution et la deuxième méthode s’appuie sur les concepts de la RLBC. En effet, plusieurs variables de décision peuvent être considérées comme des variables incrémentales, ce qui a amplement simplifié la recherche locale. Encore une fois, la stratégie de résolution a consisté à séparer la partie de la tournée et celle de l’ordonnancement (horaire des camions et chargeuses). Ainsi, la tournée des camions est contrôlé par cinq voisinages classiques largement utilisés dans les problèmes de tournées de véhicules. L’horaire des camions et des chargeuses est assuré par un algorithme glouton au niveau du modèle RLBC et occasionnellement par le solveur PPC de COMET1.1. Enfin, il est à noter que les contraintes relaxées dans ce chapitre sont les mêmes que celles relâchées auparavant au chapitre 3.

Dans le chapitre 5, nous avons présenté le SLTSP hebdomadaire qui consiste à déterminer les tournées des camions et à construire leurs horaires, en tenant compte de la synchronisation avec les chargeuses en forêts. Typiquement, les instances les plus larges que nous avons traitées comptent environ 700 demandes par semaine, une flotte de 32 camions, 5 usines et 6 sites forestiers. La dernière contribution dont fait

état le chapitre 5 reprend à quelques détails près la première phase de la méthode décrite au chapitre 4. La seule différence est que nous avons géré au niveau de cette dernière contribution les contraintes de stocks en forêts. Dans la deuxième phase qui représente les SLTSPs journaliers, nous avons intégré deux nouvelles contraintes : chaque route de camion commence et finit à la base associée au camion en question et garantit au chauffeur du camion d'avoir une pause déjeuner entre  $11h$  et  $15h$ , dans une usine. Un modèle en nombres entiers intégrant à la fois la partie tournée et horaire des camions et chargeuses a permis l'obtention d'excellents résultats. Trois explications sont derrière les performances de ce modèle. La première réside dans la structure très proche d'un problème de flot à coût minimum dans un réseau, la seconde repose sur le fait que la gestion des chargeuses est automatiquement assurée par les contraintes de conservation de flots et de capacités combinées au choix de discrétisation, ce qui entraîne que le nombre de contraintes est indépendant des lots à transporter et la troisième consiste dans le fait que les camions sont gérés implicitement dans le modèle, ce qui a permis une forte réduction de la symétrie. Une comparaison entre les deux méthodes, celles du chapitre 4 et 5 a mené à la conclusion qu'avec le dernier modèle nous obtenons de meilleurs résultats sur des instances identiques. De plus, nous avons une garantie (gap) qui mesure la qualité de la solution entière par rapport à une borne inférieure donnée par la relaxation linéaire.

## 6.1 Limites et difficultés de l'étude

Il faut dire qu'une partie importante de ce projet s'est focalisée sur la définition des contraintes de métiers associées au contexte canadien. À ce niveau, nous avons affronté plusieurs difficultés à statuer correctement toutes les contraintes et les coûts du problème. Il n'était pas très clair s'il fallait associer une base à chaque camion, puisque les camions sont la propriété de transporteurs indépendants. Toutefois, nous avons pris en considération cette contrainte dans la dernière contribution, en spéci-

fiant que chaque camion est affecté à sa base régionale.

L'accès aux données fut parfois difficile, surtout au niveau de l'information exacte du nombre de chargeuses disponibles à chaque site. Par défaut, nous avons considéré que dans chaque site et chaque usine une chargeuse est disponible. Cette hypothèse adoptée dans le premier article ne représente pas toujours la réalité. En effet, certaines forêts possèdent plus d'une chargeuse, alors que parfois plusieurs forêts n'en disposent que d'une seule. Les temps de parcours archivés par les contremaîtres incluent les pauses et les temps d'attentes. Il était compliqué de pouvoir extraire les vraies données concernant les distances, les temps d'attentes, les temps de chargement et de déchargement.

Durant tout le projet, nous avons considéré que le chargement total s'effectuait dans un seul point de chargement. Cela est vrai pour la plupart des types de bois transportés au Canada. Cependant, pour le bois de grande valeur, il n'y a pas toujours assez de quantité dans un seul site pour former un chargement total. Les approches développées au cours de cette thèse ne sont pas en mesure de traiter ce genre de problème. Nous n'avons pas, non plus, intégré à notre étude les contraintes liées aux changements et aux repos des chauffeurs. Ce qui, compte tenu des réglementations qui organisent le secteur de transport forestier, constitue un handicap.

Malheureusement, nous étions incapable d'extraire les vrais coûts des solutions manuelles fournies par FPIinnovations, pour la même raison citée auparavant, à savoir l'impossibilité de distinguer les temps de parcours des temps d'attentes et des pauses. Il faut signaler également que les contraintes supplémentaires respectées par les planificateurs des compagnies forestières (changements de chauffeurs, camions autochargeur, remplissage de carburants) ne sont pas gérées par nos approches, ce qui peut biaiser toute forme de comparaison entre nos solutions et celles de FPIinnovations. Finalement, les difficultés financières de l'industrie forestière ont obligé de leur part le partenaire avec qui nous devons effectuer la validation technique de nos

résultats à suspendre ses opérations.

## 6.2 Directions de recherches

Ce travail de recherche forme un plan de gestion du transport forestier journalier et hebdomadaire. Cependant, il serait intéressant d'intégrer les contraintes de changements de chauffeurs quand la route dépasse la limite maximale qu'un camionneur peut parcourir. Naturellement, nous pourrions nous inspirer des travaux de Flisberg et al. [2009] et de ceux d'Andersson et al. [2008] qui ont créé des noeuds de transport modélisant les changements de chauffeurs dans un réseau adapté au LTSP.

Il serait aussi pertinent de tester la relaxation lagrangienne pour résoudre le problème hebdomadaire. En effet, les contraintes de stocks et de demandes représentent les contraintes liantes de notre modèle SLTSP hebdomadaire. En relâchant ces contraintes dans l'objectif, nous obtenons un problème facile (flot à coût minimum). Nous pensons qu'une bonne stratégie gérant la mise à jour des multiplicateurs de Lagrange mènera fort probablement à des solutions entières réalisables de très bonne qualité.

En conclusion, FPIinnovations a soulevé l'intérêt des compagnies forestières pour le développement d'algorithmes et logiciels permettant de résoudre le problème du transport forestier avec horaire en temps réel. Nous prévoyons utiliser la dernière approche de cette thèse comme plan global gérant le transport forestier journalier et hebdomadaire. Par la suite, on espère élaborer des algorithmes rapides qui viendront corriger toutes les variations et imprévus susceptibles d'apparaître durant le déroulement du plan.

## BIBLIOGRAPHIE

- Andersson, G., P. Flisberg, B. Lidén, and M. Rönnqvist (2008). RuttOpt— a decision support system for routing of logging trucks. *Canadian Journal of Forest Research*, 38 : 1784–1796.
- Aykin, T (1996). Optimal shift scheduling with multiple break windows. *Management Science*, 42(4) :591–602.
- Baptiste, P., C. LePape, and W. Nuijten (1995). Constraint-Based Optimization and Approximation for Job-shop Scheduling. In *Proceedings of AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems*, pages 5–16, IJCAI’95, Montreal, Canada.
- Baptiste, P., C. LePape, and W. Nuijten (2001). Constraint-Based Scheduling. *Kluwer Academics Publishers*.
- Bechtolds, S., and L. Jacobs (1990). Implicit optimal modeling of flexible break assignments. *Management Science*, 36 : 1339–1351.
- Bramel, J., and D. Simchi-Levi (1997). On the effectiveness of set covering routing problem with time windows. *Operational Research*, 45(2) :295–301.
- Bredström, D., and M. Rönnqvist (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints *European Journal of Operational Research*, 191(1) :19–31.

- Bräysy, O (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS J. Comput*, 15 : 347–368.
- Carlsson, D., and M. Rönnqvist (1998). Tactical planning of forestry transportation with respect to backhauling. *Report LiTH-MAT-R-1998-13. Linköping University, Linköping, Sweden.*
- Chauhan, S., A. Martel, and S. D’amours (2008). Roll assortment optimization in a paper mill : an integer programming approach. *Computer and Operational Research*, 35(2) :614–627.
- Clarke, G., and J.R. Wright (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12 :568–581.
- Cordeau, J-F., G. Laporte, and A. Mercier (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52 :928–936.
- Cordeau, J-F., G. Desaulniers, J. Desrosiers, M.M. Solomon, and F. Soumis (2002). VRP with time windows. In Toth and D. Vigo (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, vol 9, Philadelphia, PA, 157–193.
- D’Amours, S., M. Rönnqvist, and A. Weintraub (2008). Using Operational Research for Supply Chain PLanning in the Forest Products Industry. *INFOR : Information Systems and Operational Research*, 46(4) :265–281.



- Dantzig, G.B., and J.H. Ramser (1959). The truck dispatching problem. *Management Sciences*, 6 :80–91.
- Dumas, Y., J. Desrosiers, and F. Soumis (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54 : 7–22.
- El Hachemi, N., M. Gendreau, and L-M. Rousseau (2008). Solving a Log-Truck Scheduling Problem with Constraint Programming. In *Proceeding of the Fifth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR08)*, 293–297.
- El Hachemi, N., M. Gendreau, and L-M. Rousseau (2009). A Hybrid LS/CP Approach to Solve the Weekly Log-Truck Scheduling Problem. In *Proceeding of the Sixth International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR09)*, 319–320.
- Epstein, R., R. Morales, J. Serón, and A. Weintraub (1999). Use of OR Systems in the Chilean Forest Industries *Interfaces*, (1) 29 : 7–29.
- Epstein, R., J. Karlsson, M. Rönnqvist, and A. Weintraub (2007). Harvest operational models in forestry, in Weintraub, A., Romero, C., Bjørndal, T., and Epstein, R. (eds) : *Handbook on Operations Research in Natural Resources*, Chapter 18. *Kluwer Academic Publishers, New York*.
- Everborn, P., P. Flisberg, and M. Rönnqvist (2006). Laps care- and operational

- system for staff planning of home care. *European Journal of Operational Research*, 171(3) : 962–976.
- Everborn, P., M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth (2009). Operations Research Improves Quality and Efficiency in Home Care. *Interfaces*, 39(1) : 18–34.
- Fahle, T., and M. Sellmann (2000). Constraint programming based column generation with knapsack subproblems. In *Proceeding of the International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems(CP-AI-OR'00)*, 33–43.
- Flisberg, P., and M. Rönnqvist (2007). Optimization based planning tools for routing of forwarders at harvest areas. *Canadian Journal of Forest Research*, 37 : 2153–2163.
- Flisberg, P., B. Lidén, and M. Rönnqvist (2009). A hybrid method based on linear programming and and tabu search for routing of logging trucks. *Canadian Journal of Forest Research*, 36(4) : 1122–1144.
- Freling, R., D. Huisman, and A. P. M. Wagelmans (2003). Models and algorithms for integration of vehicle and crew scheduling *Journal of Scheduling*, 6 :63–85.
- Fulkerson, D (1961). An out-of-kilter method for minimal-cost flow problems. *Journal of the Society of Industrial and Applied Mathematics*, 9 :18–27.
- Gendreau, M., G. Laporte, and J-Y. Potvin (2002). Metaheuristics for the Capa-

- citated VRP. In Toth and D. Vigo (eds.), *The Vehicle Routing Problem*, SIAM Monographs on Discrete Mathematics and Applications, vol 9, Philadelphia, PA, 129–154.
- Gingras, C., J-F. Cordeau, and G. Laporte (2007). Un algorithme de minimisation du transport à vide appliqué à l’industrie forestière. *INFOR*, vol.45, No.1, 41–47.
- Golden, B.L., T.L. Magnanti, and H. Nguyen (1977). Implementing vehicle routing algorithms. *Networks* 7 :113–148.
- Gronalt, M., and P. Hirsch (2007). Log-Truck Scheduling with Tabu Search Strategy. *Metaheuristics*, Vol 39 : 65–88.
- Haase, K., G. Desaulniers, and J. Desrosiers (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems *Transportation Science*, 35(3) :286–303.
- Hane, C., C. Barnhart, E. Johnson, R. Marsten, G. Nemhauser, and G. Sigismondi (1997). The fleet assignment problem : Solving a large-scale integer program. *Mathematical Programming* (70) :211–232.
- Hooker, J.N (2005). A Hybrid Method for the Planning and Scheduling. *Constraints*, Vol 10(4) :385–401.
- Ibaraki, T., S. Imahori, K. Nonobe, K. Sobue, T. Uno, and M. Yagiura (2008). An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics* (156) :2050–2069.

- Ioachim, J., J. Desrosiers, F. Soumis, and N. Bélanger (1999). Fleet assignment and routing with schedule synchronization constraints. *European Journal of Operational Research* (119) :75–90.
- John, D.S (1990). Local optimization and the traveling salesman problem. In : M.S. Paterson (ed.), *Automata, Languages and Programming*, Lecture Notes in Computer Science, vol. 443, Springer, Berlin, pp. 446–461.
- Jorgensen, R.M (1999). Planlægning i svensk skovbrug. *Institut for Matematisk Modellering. Danmarks Tekniske Universitet. IMM-EKS-1999-8(in Danish)*.
- Karlsson, J., M. Rönnqvist, and J. Bergström (2004). An optimization model for annual harvest planning. *Canadian Journal of Forest Research*, 34(8) : 1747–1754.
- Linnainmaa, S., J. Savalo, and O. Jokinen (1995). EPO : A knowledge based system for wood procurement management. Paper presented at the *7th Annual Conference on Artificial Intelligence*, Montreal.
- Martell, D.L (1982). A review of operational research studies in forest fire management. *Canadian Journal of Forest Research*, 12 : 119–140.
- Martell, D.L, E.A. Gunn, and A. Weintraub (1998). Forest management challenges for operational researchers. *European Journal of Operational Research*, 104 : 1–17.
- Michel, L., and P. Van Hentenryck (1997). A modeling language for local search In *Proceeding of the Third International Conference on the Principles and Practice of Constraint Programming (CP'97)*, pp. 237–251.

- Michel, L., and P. Van Hentenryck (2000). Localizer *Constraints*, 5 :41–82.
- Milano, M (2004). Constraint and integer programming. *Kluwer Academic Publishers*.
- Murphy, G (2003). Reducing trucks on the road through optimal route scheduling and shared log transport services *Southern Journal of Applied Forestry*, 27(3) :198–205.
- Palander, T., J. Väätäinen, S. Laukkanen, and J. Malinen (2003). Modeling backhauling on finnish energy-wood network using minimizing empty routes. *University of Joensuu, Finland, International Journal of Forest Engineering*, 15 :79–84.
- Palmgren, M., M. Rönnqvist, and P. Värbrand. A solution approach for log truck scheduling based on composite pricing and branch and bound. *International Transactions in Operational Research*, 10 :433–447.
- Palmgren, M., M. Rönnqvist, and P. Värbrand (2004). A near-exact method for solving the log-truck scheduling problem. *International Transactions in Operational Research*, 11 :447–464.
- Palmgren, M (2005). Optimal truck scheduling- mathematical modeling and solution by the column generation principle. *PHD Thesis, Dissertation No. 967, Linköping University*.
- Papadimitriou, C.H., and K. Steiglitz (1982). Combinatorial optimization : algorithms and complexity. *Prentice Hall, Englewood Cliffs, New Jersey*.

Potvin, J-Y., T. Kervahut, B.L. Garcia, and J-M. Rousseau (1996). The vehicle routing problem with time windows, Part 1 : tabu search. *INFORMS J. Comput*, 8 : 158–164.

*Le Canada et la foresterie-un aperçu* (2005). Available on the web : [www.nrcan.gc.ca/cfs – scf/national/what – quoi/canadaoverview/forests\\_f.html](http://www.nrcan.gc.ca/cfs-scf/national/what-quoi/canadaoverview/forests_f.html).

Rekik, M., J-F. Cordeau, and F. Soumis (2004). Using Benders decomposition to implicitly model tour scheduling. *Annals of Operations Research*, 128 : 111–133.

Régin, J-C (1996). *Generalized Arc Consistency for Global Cardinality Constraint*. In *Proceedings of AAAI/IAAI*, volume 1, pages 209–215. AAAI Press/The MIT Press.

Rönnqvist, M (2003). Optimization in forestry *Mathematics Programming*, 97 :267–284.

Rönnqvist, M., H. Sahlin, and D. Carlsson (1998). Operative planning and dispatching of forestry transportation. *Report LiTH-MAT-R-1998-18. Linköping University, Linköping, Sweden*.

Rönnqvist, M., and D. Ryan (1995). Solving truck dispatch problem in real time. In *Proceedings of the 31st Annual Conference of the Operational Research Society of New Zealand*, 31 August - 1 September 1995. Wellington, New Zealand. The Operational Research Society of New Zealand, Auckland, N-Z. pp. 165–172.

- Ropke, S., J.-F. Cordeau, and G. Laporte (2007). Models and Branch-and-Cut Algorithm for Pick-up and Delivery Problems with Time Windows. *Networks*, (49),4 :258–272.
- Rousseau, L-M., M. Gendreau, and G. Pesant (2002). Solving small VRPTWs with constraint programming based column generation. In *Proceeding of the fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR'02)*, p. 333–344.
- Rousseau, L-M., M. Gendreau, and G. Pesant (2003). The Synchronized Vehicle Dispatching Problem. In *Proceeding of Odysseus*, Palerme (Italie), 3 pages.
- Sakkout, H.E., and M. Wallace (2000). Probe backtrack search for minimal perturbation in dynamic scheduling. *Constraints*, Vol 5(4) :359–388.
- Savelsbergh, M.W.P., and M. Sol (1995). The general pickup and delivery problem. *Available on the web*.
- Sellmann, M., K. Zervoudakis, P. Stamatopoulos, and T. Fahle (2002). Crew Assignment via Constraint Programming : Integrating Column Generation and Heuristic Tree Search. *Annals of Operations Research*, 115 :207–225.
- Shen, Z., and J. Sessions (1989). Log-truck scheduling by network programming. *Forest Products Journal*, 39 : 47–50.

- Simonis, H., P. Charlier, and P. Kay (2000). Constraint handling in an integrated transportation problem. *IEEE Intelligent Systems*, Vol 15(1) :26–32.
- Solomon, M.M (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research*, 35(2) : 254–265.
- Solomon, M.M., and J. Desrosiers (1988). Time window constrained routing and scheduling problem. *Transportation Science*, 22 : 1–13.
- Taillard, E., P. Badeau, M. Gendreau, F. Guertin, and J-Y. Potvin (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31 : 170–186.
- Toth, P., and D. Vigo (2001). The Vehicle Routing Problem. *Society for industrial and Applied Mathematics*.
- Van Hentenryck, P (1989). Constraint Satisfaction in Logic Programming. *MIT Press*.
- Van Hentenryck, P (1999). The OPL Optimization Programming Language. *MIT Press*.
- Van Hentenryck, P., L. Perron, and J-F. Puget (2000). Search and Strategies in OPL. *ACM Transactions on Computational Logic*, (1),2 :285–320.
- Van Hentenryck, P., and L. Michel (2005). Constraint-Based Local Search. *MIT Press*.



- Van Hentenryck, P., and L. Michel (2006). Differentiable Invariants. In *Proceeding of the Twelfth International Conference on Principles and Practice of Constraint Programming (CP06)*, 685–689.
- Weintraub, A., R. Epstein, R. Morales, J. Seron, and P. Traverso (1996). A truck scheduling system improves efficiency in the forest industries. *Interfaces* (26),4 :1–12.
- Williamson, G., and M. Nieuwenhuis (1993). Integrated Timber allocation and transportation planning in Ireland. *Journal of Forest Engineering* 5 :7–15.
- Zweben, M., and M. Fox (1994). Intelligent Scheduling. *Morgan Kaufman*.

## ANNEXE A : MODÈLES EN COMET ET IMPLEMENTATION

Dans cet annexe, nous reprenons, en premier lieu, les modèles du chapitre 4, non pas tels quels, mais conformément au logiciel COMET. Le premier modèle planifie les voyages pleins de la semaine, alors que le deuxième construit la tournée et l'horaire des camions.

### Modèle tactique

```

Solver<LS> m();

var {bool} openfj(m) ∀j ∈ J, ∀f ∈ F

var {int} stockwpj(m, 0..Smax) ∀j ∈ J, ∀w ∈ W, ∀p ∈ P

var {int} tripfwpj(m, 0..Tmax) ∀j ∈ J, ∀f ∈ F, ∀w ∈ W, ∀p ∈ P

FunctionSum<LS> OS(m);

OS.post(C*openfj) ∀j ∈ J ∀f ∈ F

+OS.post(cfwptripfwpj) ∀j ∈ J, ∀f ∈ F, ∀w ∈ W, ∀p ∈ P

ConstraintSystem<LS> Sys(m);

Sys.post(stockwpj ≤ Spmax) ∀j ∈ J, ∀w ∈ W, ∀p ∈ P

```

$\text{Sys.post}(\text{stock}_{wp}^{j-1} + \sum_{f \in F} \text{trip}_{fwp}^j = D_{wp}^j + \text{stock}_{wp}^j) \forall j \in J, \forall w \in W, \forall p \in P$   
 $\text{Sys.post}(\sum_{w \in W} \sum_{p \in P} \text{trip}_{fwp}^j \leq U.\text{open}_f^j) \forall j \in J, \forall f \in F$   
 $\text{Sys.post}(\sum_{w \in W} \sum_{p \in P} \text{trip}_{fwp}^j \geq L.\text{open}_f^j) \forall j \in J, \forall f \in F$   
 $\text{m.close}();$

### Modèle journalier

$\text{Solver}<\text{LS}> \text{m}();$   
 $\text{var } \{int\} \text{Start}_{P_r}(m, 0..T) \forall r \in R$   
 $\text{var } \{int\} \text{Start}_{D_r}(m, 0..T) \forall r \in R$   
 $\text{var } \{int\} \text{Prec}_r(m, I) \forall r \in O$   
 $\text{var } \{int\} \text{End}_{P_r}(m, 0..T) \leftarrow \text{Start}_{P_r} + d_p \forall r \in R$   
 $\text{var } \{int\} \text{End}_{D_r}(m, 0..T) \leftarrow \text{Start}_{D_r} + d_d \forall r \in R$   
 $\text{var } \{int\} \text{Wait}_{M_r}(m, 0..T) \leftarrow \text{Start}_{D_r} - D[M_r, F_r] - \text{End}_{P_r} \forall r \in R$   
 $\text{var } \{int\} \text{Wait}_{F_r}(m, 0..T) \leftarrow \text{Start}_{P_r} - D[M_{Prec_r}, F_r] - \text{End}_{D_{Prec_r}} \forall r \in R$   
 $\text{var } \{int\} \text{Start}_{F_f}(m, 0..T) \leftarrow \min(\text{Start}_{P_r}), \forall r \in R : F_r = f$   
 $\text{var } \{int\} \text{End}_{F_f}(m, 0..T) \leftarrow \max(\text{End}_{P_r}), \forall r \in R : F_r = f$   
 $\text{FunctionSum}<\text{LS}> \text{OS(m)};$   
 $\text{OS.post}(c_d D[M_{Prec_r}, F_r] + c_t (\text{Wait}_{M_r} + \text{Wait}_{F_r})) \forall r \in R$   
 $+ \text{OS.post}(c_l (\text{End}_{F_f} - \text{Start}_{F_f})) \forall f \in F$

```

ConstraintSystem<LS> Sys(m);

Sys.post(|StartPr1 - StartPr2| ≥ dp), ∀r1, r2 ∈ R : Fr1 = Fr2

Sys.post(|StartDr1 - StartDr2| ≥ dd), ∀r1, r2 ∈ R : Mr1 = Mr2

Sys.post(EndDr ≤ T) ∀r ∈ R

Sys.post(ALLDIFFERENT(Prec))

Sys.post(Precr ≠ r), ∀r ∈ R

Sys.post(WaitMr ≥ 0), ∀r ∈ R

Sys.post(WaitFr ≥ 0), ∀r ∈ R

m.close();

```

En second lieu, nous présentons le code, en COMET, de certaines fonctions tels l'algorithme glouton, la fonction qui retrouve le plus court chemin entre deux noeuds, etc.

### **Fonction plus court chemin entre deux noeuds**

```

function int[] Dijk(int noeud_depart, int[,] dist, int noeud_arrive)
{
    int test = 1;

    int dim = dist.sz(0);

    range vy = 0..(dim-1);

    set{int} S[1..1];

```

```

    forall(i in vy: i != noeud_depart)
    {
        S[1].insert(i);
    }
    \\ Tous les noeuds du graphe mise à part
    le noeud de départ choisi sont
    insérés dans l'ensemble S
    }

    range VY = 0..dim;

    int chemin[vy,vy] = -1;

    forall(i in vy)
    {
        chemin[i,0] = noeud_depart;
    }

    int pi[vy] = 0;

    forall(x in vy: x != noeud_depart)
    {
        pi[x] = dist[noeud_depart,x];
    }
    \\ Calcul la distance entre le noeud
    de départ choisi et n'importe quel
    autre noeud du graphe
    }

    while(test > 0)
    {
        if(S[1].getSize() > 0)
        {
            selectMin(x in S[1])(pi[x])
            \\ On sélectionne le noeud le
            plus proche du noeud de départ
            appartenant à S (on utilise selectMin de COMET)
            {
                S[1].delete(x); \\ On le retire de l'ensemble S
                int position = 0;
                while(chemin[x,position] != -1)

```

```

    {
        position++;
    }

    chemin[x,position] = x;

    forall(y in S[1])
    {
        if(pi[y] > pi[x] + dist[x,y])
        {
            pi[y] = pi[x] + dist[x,y];
            \\ Pour chaque noeud y,
                on met à jour la plus petite
                distance le séparant du noeud de départ

            forall(k in vy)
            {
                chemin[y,k] = chemin[x,k];
            }
            \\ Retracer le plus court
            chemin à date entre le
            noeud de départ et le noeud y
        }

    }

}

}

else
{
    test = 0;
}

}

int theway[VY] = 0;

```

```

\\ On initialise le plus court
    chemin entre le noeud de départ et celui d'arrivée

forall(k in vy)
{
    theway[k] = chemin[noeud_arrive,k];
\\ Le plus court chemin entre le noeud de départ
    et le noeud d'arrivée
}
theway[dim] = pi[noeud_arrive];
\\ À la fin du plus court chemin, on met la valeur de la
    de la distance séparant les deux noeuds
return theway;

}

```

## Heuristique gloutonne d'horaire de camions

```

function int[,] greedy_schedule(int[,] lot, int[] Si, int[] Us,
    int nv, int nbT, int _nbS, int _nbU, int[,] Dt,
    int[] Udepart, int time_c, int time_d)
{
    int NBT = nbT;
    int nbS = _nbS;
    int nbU = _nbU;
    range sites = 0..(nbS-1);
    range usines = 0..(nbU-1);
    range S = 0..nbS;
    range U = 0..nbU;
    range veh = 0..(nv-1);
    range VEH = 0..(2*nv-1);
    range activites = 0..(NBT-1);
    \\ On initialise tous les paramètres dont on a besoin

    int St_C[veh,activites] = -1;
    int St_D[veh,activites] = -1;
    \\ On initialise le début des activités de chargement
    et déchargement associées aux camions
    int schedule[VEH,activites] = 0;
    \\ On initialise l'horaire que retourne cette fonction
    int ctr[veh] = 0;
    \\ Un compteur par camion permettant de traiter dans
    l'ordre les requêtes de chaque camion
    int dly[veh] = 0;
    \\ Un délai par camion permettant de retarder
    l'horaire de chaque activité (chargement ou
    déchargement) lorsque la chargeuse associée
    n'est pas disponible

    int nbact = 1;
    int horizon = 1000;
    range Horizon = 0..horizon;
    \\ L'horizon de l'horaire
    int Sbusy[S,Horizon] = 0;
    int Ubusy[U,Horizon] = 0;
    \\ Deux matrices retournant l'état (disponible ou occupée)
    de chaque chargeuse en forêt ou usine

```



```

forall(i in veh)
{
  int k = 0;

  while(lot[i,k] >= 1 && lot[i,k] <= nbT)
  {
    if(k >=1)
    {
      St_C[i,k] = Dt[Us[lot[i,k-1]],Si[lot[i,k]]]+St_D[i,k-1]+time_d;
      St_D[i,k] = St_C[i,k] + time_c + Dt[Us[lot[i,k]],Si[lot[i,k]]];
      k++;
    }
    \\ On initialise le début de chaque chargement
    et déchargement sans tenir compte des chargeuses
  else
  {
    St_C[i,k] = Dt[Usdepart[i],Si[lot[i,k]]];
    St_D[i,k] = St_C[i,k]+time_c+Dt[Us[lot[i,k]],Si[lot[i,k]]];
    k++;
  }

}

}

while(nbact <= NBT)
{
  selectMin(i in veh: St_C[i,ctr[i]] >= 0 )(St_C[i,ctr[i]] + dly[i])
  {
    \\ On sélectionne le camion ayant le plus petit temps de début
    de chargement non encore traité
    while(Sbusy[Si[lot[i,ctr[i]]],St_C[i,ctr[i]] + dly[i]] == 1)
    {
      dly[i] = dly[i] + 1;
    }
    \\ On retarde le début de chaque chargement si
    la chargeuse associée est occupée

    Sbusy[Si[lot[i,ctr[i]]],St_C[i,ctr[i]] + dly[i]] = 1;
    St_C[i,ctr[i]] = St_C[i,ctr[i]] + dly[i];
  }
}

```

```

while(Ubusy[Us[lot[i,ctr[i]]],St_D[i,ctr[i]] + dly[i]] == 1)
{
    dly[i] = dly[i] + 1;
}
\\ On retarde le début de chaque déchargement si
    la chargeuse associée est occupée

Ubusy[Us[lot[i,ctr[i]]],St_D[i,ctr[i]] + dly[i]] = 1;
St_D[i,ctr[i]] = St_D[i,ctr[i]] + dly[i];
ctr[i] = ctr[i] + 1;
}
    nbact++;
}

forall(i in veh)
{
    forall(k in activites)
    {
        schedule[i,k] = St_C[i,k];
        schedule[i+nv,k] = St_D[i,k];
    }
}
\\ On retourne l'horaire de manière à ce que les premières (nv)
    lignes représentent le temps de début des chargements,
    alors que les dernières (nv) lignes représentent le temps
    de début des déchargements

return schedule;
}

```